ANL488 FINAL PROJECT REPORT

Predictive Modelling on Employees' Absenteeism at Work

Submitted by
Goh Ya Qi, Ennice

SCHOOL OF BUSINESS
Singapore University of Social Sciences

Presented to Singapore University of Social Sciences
in partial fulfilment of the requirements for the
Degree of Bachelor of Science
in Business Analytics

Name	Goh Ya Qi, Ennice
Course Code	ANL488
Title of TMA	**Understanding Employee Attrition: An Explainable AI Approach with SHAP and XGBoost (Project Final Report)**
SUSS PI No.	Z1970248
Submission Date	5 November 2023

Table of Contents

Abstract

Employee attrition represents a significant challenge for organizations worldwide, not only due to the substantial costs incurred but also because of the prevailing lack of understanding of its fundamental causes. The business objective is to understand the key factors contributing to employee attrition in order to provide a data-driven approach to improving employee retention. This study leverages a publicly available IBM HR dataset sourced from Kaggle, extending beyond previous research by adopting an Explainable AI (XAI) approach with the Shapley Additive exPlanations (SHAP) framework. The XGBoost Classifier model achieves robust test ROC-AUC scores exceeding 99%. SHAP yields a nuanced, in-depth understanding of the principal factors influencing employee attrition, which is used to develop strategic, data-driven recommendations. Business users and stakeholders can access the deployed model through an interactive Tableau dashboard, enabling them to craft their own strategies based on the conducted analysis.

Chapter 1: Introduction

Employee attrition is a complex challenge that organizations are increasingly grappling with, a trend that has been exacerbated in recent years. A Gartner study stated that companies should expect a 50-75% higher year-over-year attrition rate than before, with the amount of time needed to hire a suitable employee increasing by 18% since before the pandemic. (Wiles, 2021). Such dramatic shifts in the employment landscape have far-reaching consequences for businesses, both in terms of direct expenses and intangible costs (Moon et al., 2022).

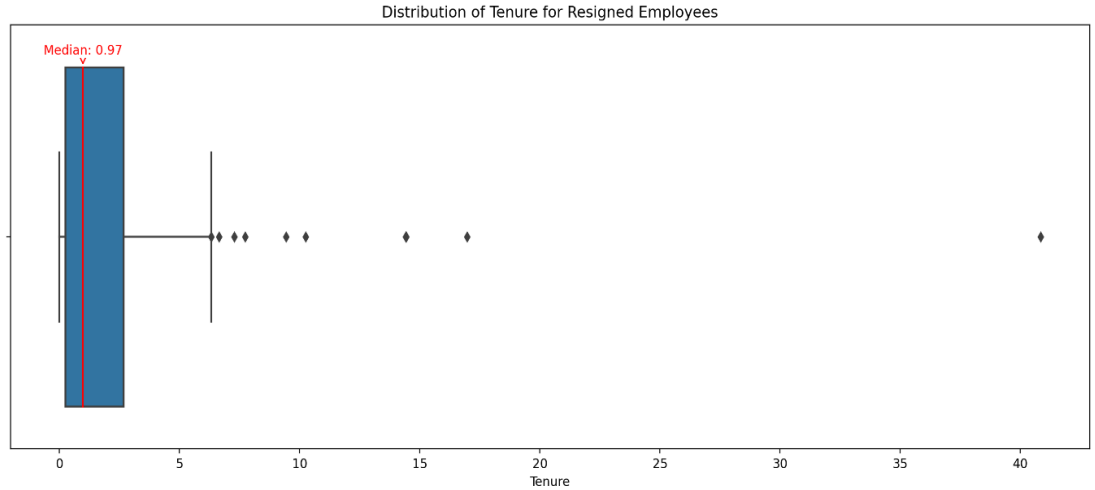
From a financial standpoint, employee attrition represents a significant challenge for industries across the board. As noted by the Society for Human Resource Management (SHRM, 2016), the direct replacement costs due to turnover can consume as much as 50%-60% of an employee's annual salary, with the overall costs potentially soaring to 90%-200% when accounting for the indirect repercussions, such as lost productivity and training expenses for new hires.

These financial implications are drawing increased attention from companies, which recognize the urgent need to understand and address the root causes of employee attrition. Despite this awareness, many organizations find themselves at a loss to pinpoint the exact factors driving this trend. A report by De Smet et al. (2021) highlights this gap in understanding, emphasizing that employers must delve into the specific reasons behind employee departures if they are to effectively mitigate attrition and its accompanying financial strain.

The Attrition Challenge at Winner Engineering Pte Ltd

This study is conducted on behalf of Winner Engineering Pte Ltd. Winner Engineering is a small and medium-sized enterprise (SME) focusing on commercial air-conditioning and mechanical ventilation solutions. It has a workforce of just over 90 full-time employees. It should be noted that this figure only includes full-time employed staff and does not account for construction workers, who are engaged on a contractual basis and are not entitled to the same employment benefits.

Based on data provided by Winner Engineering's HR department (visualized in Figure 1), about half of its staff leave within the first year of their employment, and alarmingly, about a quarter depart within the first 3 months. This turnover rate stands in stark contrast to the construction industry's median employee tenure, which was 3.9 years in 2022. (Bureau of Labour Statistics, 2022).




```
Q1: 0.24
Median: 0.97
Q3: 2.67
```

Figure 1: Boxplot of Employee Tenure at Winner Engineering

Despite dedicated efforts to improve employment benefits and foster an employee-centric workplace culture in recent years, Winner Engineering continues to face high turnover rates. These persistent rates of attrition have adversely impacted productivity, escalated costs, and impacted employee morale.

In response to these challenges, this study adopts a predictive modelling approach, in conjunction with Explainable AI (XAI) techniques, to understand and address the factors contributing to employee attrition at Winner Engineering. Due to the small size of the company's workforce, the volume of HR employee data available was insufficient for building a reliable predictive model, as it lacked the quantity of data points needed to reliably and accurately capture patterns in the data. Therefore, this research utilizes the 'IBM HR w/ more rows' dataset obtained from Kaggle. (Kaggle, 2017). This dataset contains an extensive range of employee attributes, such as demographics, job roles, satisfaction levels, and other relevant workplace metrics. These attributes are vital in constructing a reliable predictive model.

Although there is already existing research that use similar publicly available HR datasets, this study is unique as it employs explainable AI (XAI) techniques for model explanations and interpretations. We utilise SHAP, an XAI framework which details how features contribute to model outcomes, identifies feature interactions, and uncovers trends within the model's decision-making process. This allows us to gain clear, actionable insights that inform business decision-making.

In line with this, the business objective of this study is to understand the key factors contributing to employee attrition, with the aim of crafting a data-driven and strategic

approach to talent management, hiring, and employee retention. Complementing this, the data mining objective is to accurately predict employee attrition using a wide array of employee demographics and job-related factors.

This study is presented in ten structured chapters. Chapter 1 introduced the study. Chapter 2 will explore relevant literature on employee attrition and various methodologies, and Chapter 3 will discuss data understanding and preparation. Chapter 4 describes the predictive modelling approach, while Chapter 5 evaluates model outcomes. Insights from model interpretations are discussed in Chapter 6, followed by actionable recommendations in Chapter 7. Chapter 8 considers the study's limitations and potential future research. Model deployment through a Tableau interactive dashboard is discussed in Chapter 9. Chapter 10 concludes this study with a brief summary on the key findings and overarching conclusions.

Chapter 2: Literature Review

Employee attrition refers to the process of employees leaving the company voluntarily or involuntarily. (Punnoose & Ajit, 2016). The study of attrition has garnered much attention from researchers and organisations due to the significant costs -- both direct and indirect -- associated with it. Direct costs include recruitment expenses, productivity loss, the labour gap before a replacement is onboarded, and the initial reduced efficiency of new recruits. (Cascio & Boudreau, 2008). Employee departures also lead to the loss of resources invested in their training and development. (Mello, 2011). In addition, Boushey and Glynn (2012) emphasised the indirect costs, highlighting the negative impact on organisational performance and the erosion of knowledge and social capital. Furthermore, high turnover rates can undermine the morale and sense of job security of the remaining workforce, potentially leading to further resignations. (Hancock et al., 2013).

Many studies have been conducted to examine which factors correlate the most to employee attrition. In a meta-analysis of studies on employee turnover, the main factors identified were age, tenure, salary, job satisfaction, and the employee's sense of fairness. (Cotton & Tuttle, 1986). Furthermore, demographic features such as age, gender, ethnicity, education, and marital status have been noted to influence employee attrition (Peterson, 2004). Moreover, a study conducted by Saeed et al. (2023) emphasized the influence of workplace-related factors, notably salary and job satisfaction, on attrition.

Leveraging SHAP for model explainability

While the studies above have detailed various models to examine attrition and its factors, they do not detail the impact of each factor on the model's prediction beyond a statistical outcome. In contrast, a recent study by Mohiuddin et al. (2023) demonstrated a system which uses explainable AI library SHAP (Shapley Additive exPlanations) to explain and quantify the impact of the key factors affecting attrition.

SHAP is a model-agnostic, unified method to measure the importance and impact of individual features on the predictions of machine learning models. It promotes transparency and interpretability in "black-box" machine learning models which are inherently difficult to explain due to their complexity. For example, the system designed by Mohiuddin et al. (2023) relied on an extreme gradient boosting (XGBoost) model, a highly accurate and complex model that is difficult to interpret with traditional statistical methods. SHAP offers a comprehensive suite of metrics that facilitate both global and local interpretability, which provides insights into the model's overall decision-making patterns down to the individual prediction. Furthermore, a study by Lundberg & Lee (2017) found that there was a strong

alignment between human explanation and SHAP results. In the Model Interpretation section of this study, we will use SHAP to identify the most influential factors and to examine their impact on employee attrition.

Machine learning models

Much research has been done into various predictive modelling techniques and approaches, seeking to improve the outcomes of human resource management. (Yue et. al., 2018). Chen (2023) proposed a logistic regression approach that determines the importance of each predictor based on its statistical significance, which is based on the p-value of each feature. While this method offers highly interpretable results, its accuracy is limited as only 15% of actual attriters were correctly predicted.

In a study by Zhao et al. (2019) comparing various tree-based and non-tree based classifiers for attrition prediction, ensemble tree-based methods, particularly Extreme Gradient Boosting (XGBoost) and Gradient Boosting Trees (GBT), demonstrated superior performance. These ensemble gradient boosting approaches were remarkably stable and showed high predictive ability, with median ROC scores surpassing 94% across multiple datasets. Their performance far surpasses that of K-nearest neighbours, naive bayes, support vector machine and logistic regression, which had median ROC scores in the 70-80% range and far higher variance in scores across datasets. Punnoose and Ajit (2016) reported comparable findings in their research using a separate HR dataset.

In this study, we will use logistic regression as a baseline model and adopt XGBoost for final predictions should it demonstrate significant improvement in performance over the baseline

model.

Dealing with imbalanced classes

A common problem faced in classification problems is that of imbalanced class distribution. This issue occurs frequently in attrition datasets where the number of positive samples is significantly lower than the number of negative samples because there are fewer attriters than non-attriters. When training a machine learning model on imbalanced datasets, there's a propensity for the model to favour the majority class, resulting in accurate predictions for non-attriters but poor performance in identifying attriters.

This issue is evident in Chen (2023)'s study, where the model exhibited a high accuracy of 99.4% for predicting non-attriters but only a 15.2% accuracy for identifying attriters. Alduayj and Rajpoot (2018) found that synthetically oversampling the minority class using SMOTE resulted in a noticeable improvement in model performance. SMOTE generates new instances by performing linear interpolations between a chosen instance and its nearest neighbours. (Zhu et al., 2017). Many studies have highlighted the effectiveness of SMOTE in addressing imbalanced datasets, with other oversampling methods showing minimal performance variations in comparison. (Kovacs, 2019).

Wei et al. (2023) further demonstrated that simultaneous tuning of SMOTE hyper-parameters and model hyper-parameters during training significantly increases model performance. Building on these insights, our study will use SMOTE to amplify the minority class (attriters) and will include hyper-parameter tuning for SMOTE during model training.

Chapter 3: Data Understanding and Data Preparation

I. Dataset Introduction and Data Cleaning

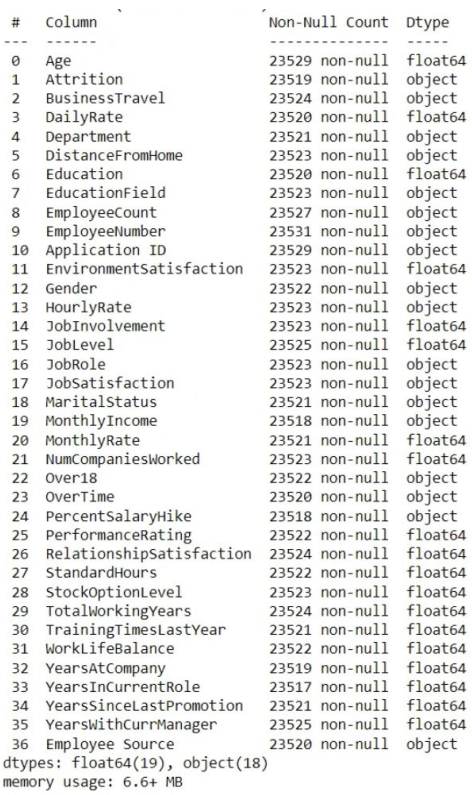
The "IBM HR w/ more rows" dataset, curated and provided by IBM data analysts, delves into HR analytics to understand employee attrition and performance. It encompasses many features, ranging from employee demographic features such as age and education to job-specific features like job satisfaction and monthly income. The original dataset contains 23 532 rows and 37 columns. The data dictionary is found in Appendix A.

A snapshot of the data can be found in Figure 1, and the automatically assigned data types can be found in Figure 2.

Upon initial examination, several columns appear to have names indicating potential overlap or high correlation. Notably, MonthlyIncome and MonthlyRate present such a possibility, as do YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, and YearsWithCurrManager. This will be examined in the Correlation analysis section later in this chapter.

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	StandardH
0	41.000	Voluntary Resignation	Travel_Rarely	1102.000	Sales	1	2.000	Life Sciences	1	1	...	80
1	37.000	Voluntary Resignation	Travel_Rarely	807.000	Human Resources	6	4.000	Human Resources	1	1	...	80
2	41.000	Voluntary Resignation	Travel_Rarely	1102.000	Sales	1	2.000	Life Sciences	1	1	...	80
3	37.000	Voluntary Resignation	Travel_Rarely	807.000	Human Resources	6	4.000	Marketing	1	4	...	80
4	37.000	Voluntary Resignation	Travel_Rarely	807.000	Human Resources	6	4.000	Human Resources	1	5	...	80
5	37.000	Voluntary Resignation	Travel_Rarely	807.000	Human Resources	6	4.000	Marketing	1	6	...	80
6	41.000	Voluntary Resignation	Travel_Rarely	1102.000	Sales	1	2.000	Life Sciences	1	7	...	80
7	41.000	Voluntary Resignation	Travel_Rarely	1102.000	Sales	1	2.000	Life Sciences	1	8	...	80
8	41.000	Voluntary Resignation	Travel_Rarely	1102.000	Sales	1	2.000	Life Sciences	1	9	...	80
9	41.000	Voluntary Resignation	Travel_Rarely	1102.000	Sales	1	2.000	Life Sciences	1	10	...	80

Figure 1: Snapshot of the raw data (only first few columns and rows shown)

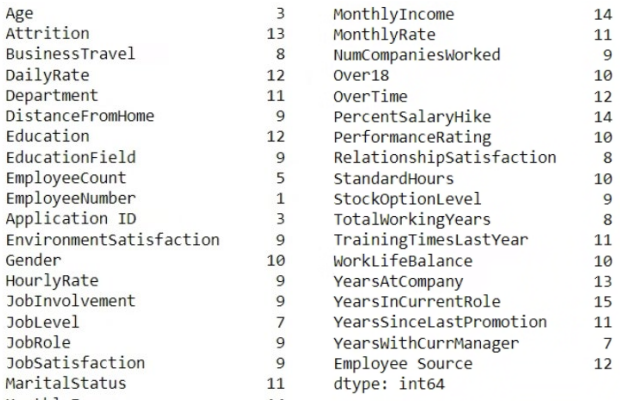


#	Column	Non-Null Count	Dtype
0	Age	23529 non-null	float64
1	Attrition	23519 non-null	object
2	BusinessTravel	23524 non-null	object
3	DailyRate	23520 non-null	float64
4	Department	23521 non-null	object
5	DistanceFromHome	23523 non-null	object
6	Education	23520 non-null	float64
7	EducationField	23523 non-null	object
8	EmployeeCount	23527 non-null	object
9	EmployeeNumber	23531 non-null	object
10	Application ID	23529 non-null	object
11	EnvironmentSatisfaction	23523 non-null	float64
12	Gender	23522 non-null	object
13	HourlyRate	23523 non-null	object
14	JobInvolvement	23523 non-null	float64
15	JobLevel	23525 non-null	float64
16	JobRole	23523 non-null	object
17	JobSatisfaction	23523 non-null	object
18	MaritalStatus	23521 non-null	object
19	MonthlyIncome	23518 non-null	object
20	MonthlyRate	23521 non-null	float64
21	NumCompaniesWorked	23523 non-null	float64
22	Over18	23522 non-null	object
23	OverTime	23520 non-null	object
24	PercentSalaryHike	23518 non-null	object
25	PerformanceRating	23522 non-null	float64
26	RelationshipSatisfaction	23524 non-null	float64
27	StandardHours	23522 non-null	float64
28	StockOptionLevel	23523 non-null	float64
29	TotalWorkingYears	23524 non-null	float64
30	TrainingTimesLastYear	23521 non-null	float64
31	WorkLifeBalance	23522 non-null	float64
32	YearsAtCompany	23519 non-null	float64
33	YearsInCurrentRole	23517 non-null	float64
34	YearsSinceLastPromotion	23521 non-null	float64
35	YearsWithCurrManager	23525 non-null	float64
36	Employee Source	23520 non-null	object

dtypes: float64(19), object(18)
memory usage: 6.6+ MB

Figure 2: Data types automatically assigned to all columns in dataset

Data cleaning was performed using the pandas library. While several columns contained missing values, only 233 rows out of the total 23,532 had these discrepancies. Given the small proportion of affected rows, any row containing missing values was removed.



Age	3	MonthlyIncome	14
Attrition	13	MonthlyRate	11
BusinessTravel	8	NumCompaniesWorked	9
DailyRate	12	Over18	10
Department	11	OverTime	12
DistanceFromHome	9	PercentSalaryHike	14
Education	12	PerformanceRating	10
EducationField	9	RelationshipSatisfaction	8
EmployeeCount	5	StandardHours	10
EmployeeNumber	1	StockOptionLevel	9
Application ID	3	TotalWorkingYears	8
EnvironmentSatisfaction	9	TrainingTimesLastYear	11
Gender	10	WorkLifeBalance	10
HourlyRate	9	YearsAtCompany	13
JobInvolvement	9	YearsInCurrentRole	15
JobLevel	7	YearsSinceLastPromotion	11
JobRole	9	YearsWithCurrManager	7
JobSatisfaction	9	Employee Source	12
MaritalStatus	11	dtype: int64	

Figure 3: Rows with missing values in the dataset

Subsequently, 14 sets of duplicate rows were identified, all displaying consistent Employee Numbers and column values. This indicates that they represent the same employee. To maintain data integrity, only the first instance of these duplicates will be retained, and the rest will be removed.

Afterward, unique identifier columns like EmployeeNumber and Application ID are removed. Columns like Over18, StandardHours and EmployeeCount have the same value throughout the dataset. These columns do not contribute valuable information to the predictive model and may introduce undesired complexity, so they are removed.

Some discrepancies were noticed where two separate rows in the Employee Source and EducationField columns had the value "Test". It is likely that these entries represent test data rather than authentic records, and as such, they are removed.

There are incorrectly recognised data types for some columns identified. For instance, the DistanceFromHome column, which should inherently be numeric, was identified as a string.

Such columns are converted to the appropriate numeric data type. Moreover, numeric columns containing integer values but identified as float64 were converted to int32 data type.

II. Exploratory data analysis

The aim of exploratory data analysis is to identify factors that may impact an employee's tenure and probability to leave the company. Through this process, we aim to gain meaningful insights that can guide subsequent analyses and model development.

It is worth noting that the exploratory data analysis is done on the IBM HR dataset and reflects the

Figure 4: Distribution of Years at Company

The variable "Years at Company" is a continuous variable that represents how long employees have stayed at the company. This variable is right skewed and has a median of 5 years. This indicates that half of the employees leave in under 5 years of working for the company.

Therefore, the variable "Attrition" is revised by transforming it into two nominal categories, "Attrition and "Non-Attrition". The median tenure of 5 years provides a good cut-off point for categorisation. We categorise employees who have resigned and have less than 5 years' tenure as Attrition and the rest as Non-Attrition.

```
6
7  # Set rows that match the condition to 'Attrition'
8  df.loc[(df['Attrition'] == 'Voluntary Resignation') &
9         (df['YearsAtCompany'] < 5), 'Attrition'] = 'Attrition'
10
11 # Set all other rows to 'Non-Attrition'
12 df.loc[df['Attrition'] != 'Attrition', 'Attrition'] = 'Non-Attrition'
```

Figure 5: Transformation of the target variable

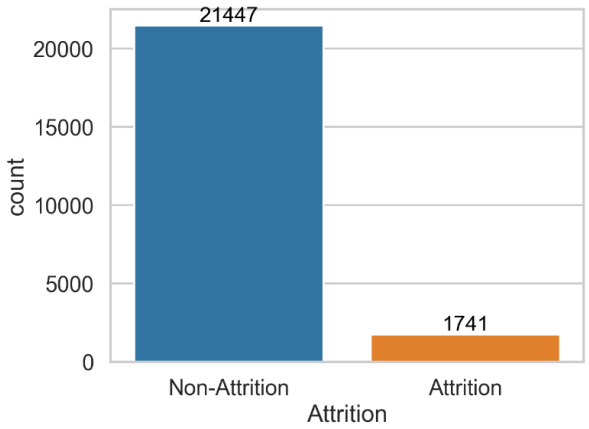


Figure 6: Distribution of Non-Attrition to Attrition

The data shows a pronounced class imbalance with 21,000 employees under Non-Attrition compared to just 1,741 in the Attrition category. As such, machine learning models are very likely to be biased towards the majority class, Non-Attrition, because they aim to minimize overall error. The modelling section will discuss several techniques used to address this.

We will examine some of the features and their relationship with the target variable.

Employee demographic factors

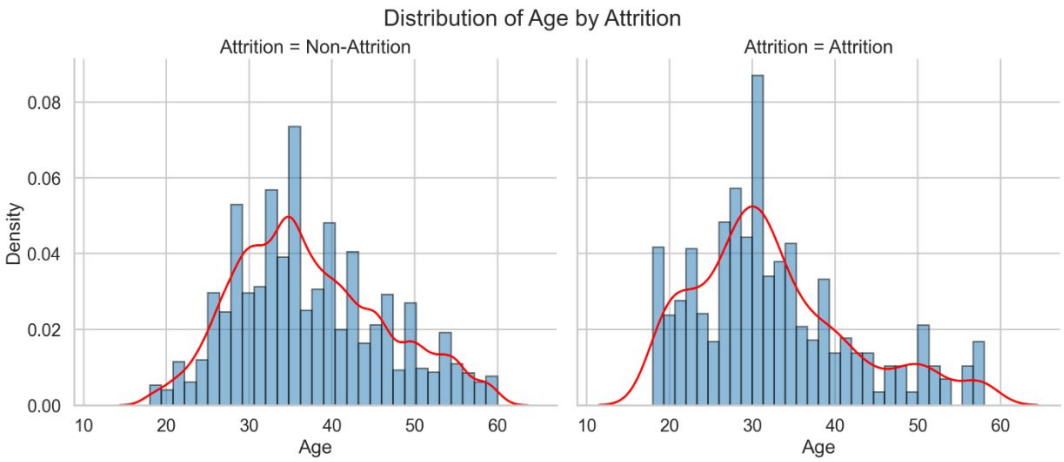


Figure 7: Distribution of Age by Attrition

In the density plot, the spread for non-attrition is slightly wider than the spread for attrition. In addition, the curve for non-attrition peaks at around 35 years while the curve for attrition peaks at around 30 years. The data suggests that individuals, particularly those around the age of 30, have a marginally higher propensity to depart from the company. This observation aligns with the findings of Chen (2023), who similarly identified a slight negative correlation between age and propensity of attrition in their research.

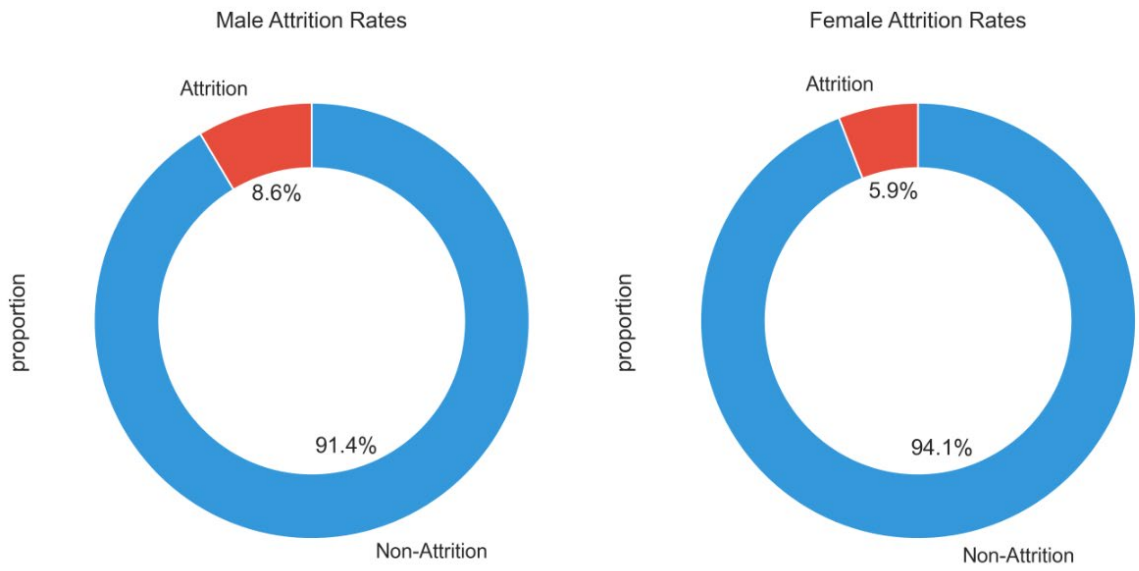


Figure 9: Employee Attrition by Gender

The donut charts depict attrition rates based on gender. Males exhibit an attrition rate of 8.6%, with a substantial 91.4% non-attrition. In comparison, females have a slightly lower attrition rate of 5.9%, accompanied by a higher non-attrition rate of 94.1%. This difference suggests that gender may have some impact on attrition and males are more likely to leave their positions than females.

Job-related factors

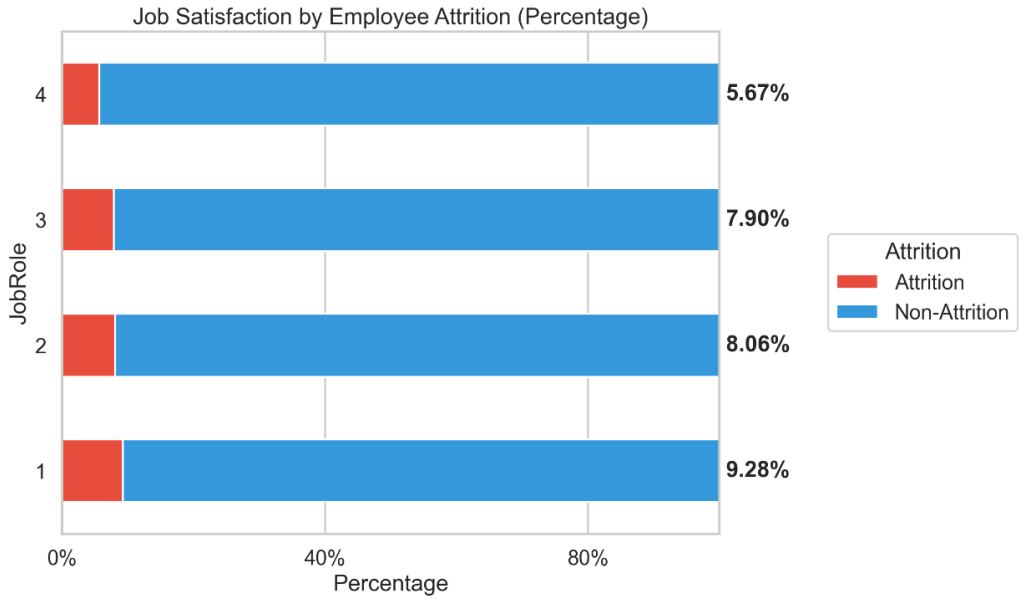


Figure 8: Employee Attrition by Job Satisfaction

The 100% stacked bar plot reveals an inverse relationship between job satisfaction and attrition rates. As job satisfaction ratings increase from "Very Low" (1) to "High" (4), attrition rates correspondingly decrease, with the highest satisfaction level witnessing the lowest attrition at 5.7%. This suggests that job satisfaction could be a factor influencing employee retention, as found by Saeed et al. (2023). Chen's (2023) research also demonstrates an inverse relationship between job satisfaction and attrition propensity, further corroborating this point.

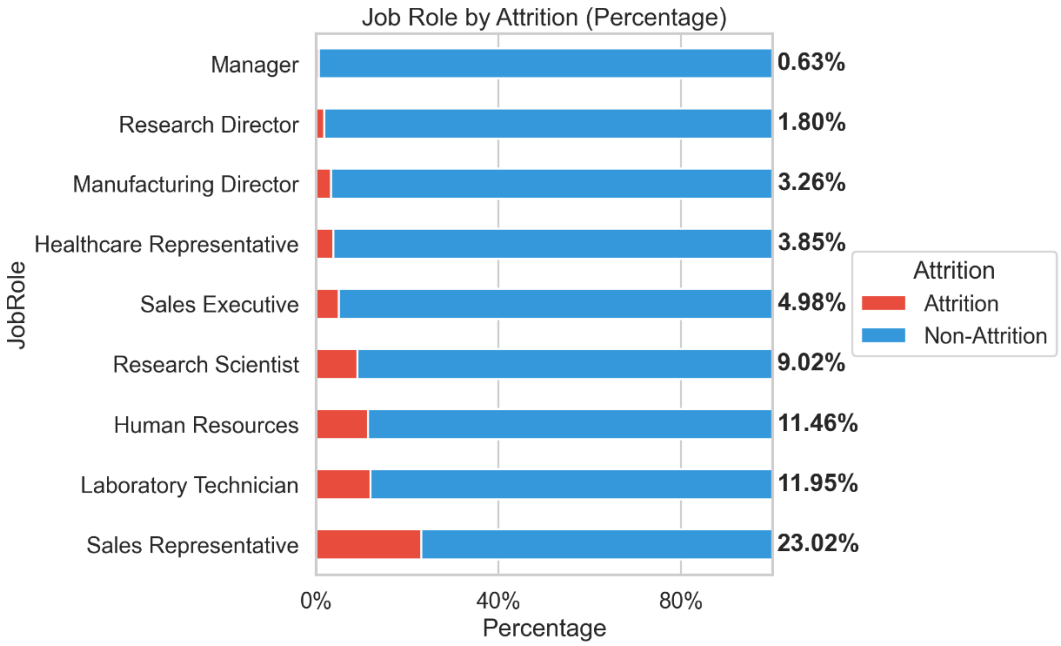


Figure 9: Job Role by Attrition

The 100% stacked bar plot illustrates the variance in attrition rates among different job roles. A significant observation is that Sales Representatives experience the highest attrition rate at 23.02%, while Managers have the lowest at 0.63%. Roles such as Human Resources and Laboratory Technicians also demonstrate relatively high attrition rates.

While the specific roles and their attrition rates may vary based on the industry and the nature of individual companies, this data highlights that the percentage of employees leaving can significantly differ by job roles.

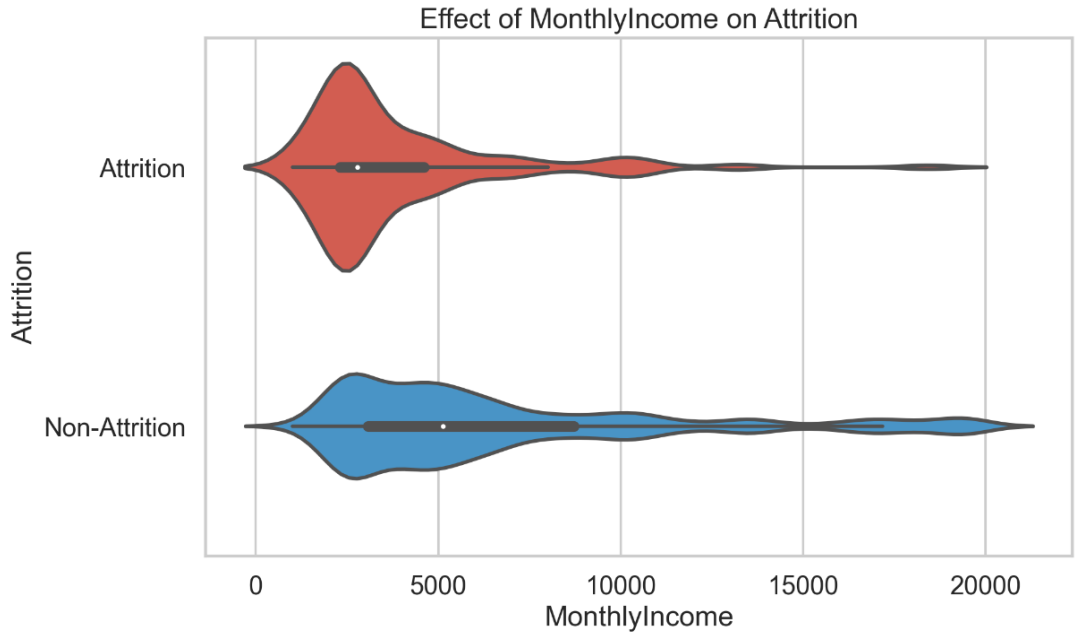


Figure 10: Monthly Income by Attrition

The violin plot shows that most of the employees who attrit tend to be of lower income roles, compared to those who remain in the company. It can be observed that most employees who attrit fall in the lower monthly income bracket, while employees who stay have a more consistent representation across varying income levels. This implies that monthly income plays a role in an employee's decision to leave or stay within the company, a point emphasized by Saeed et al. (2023).

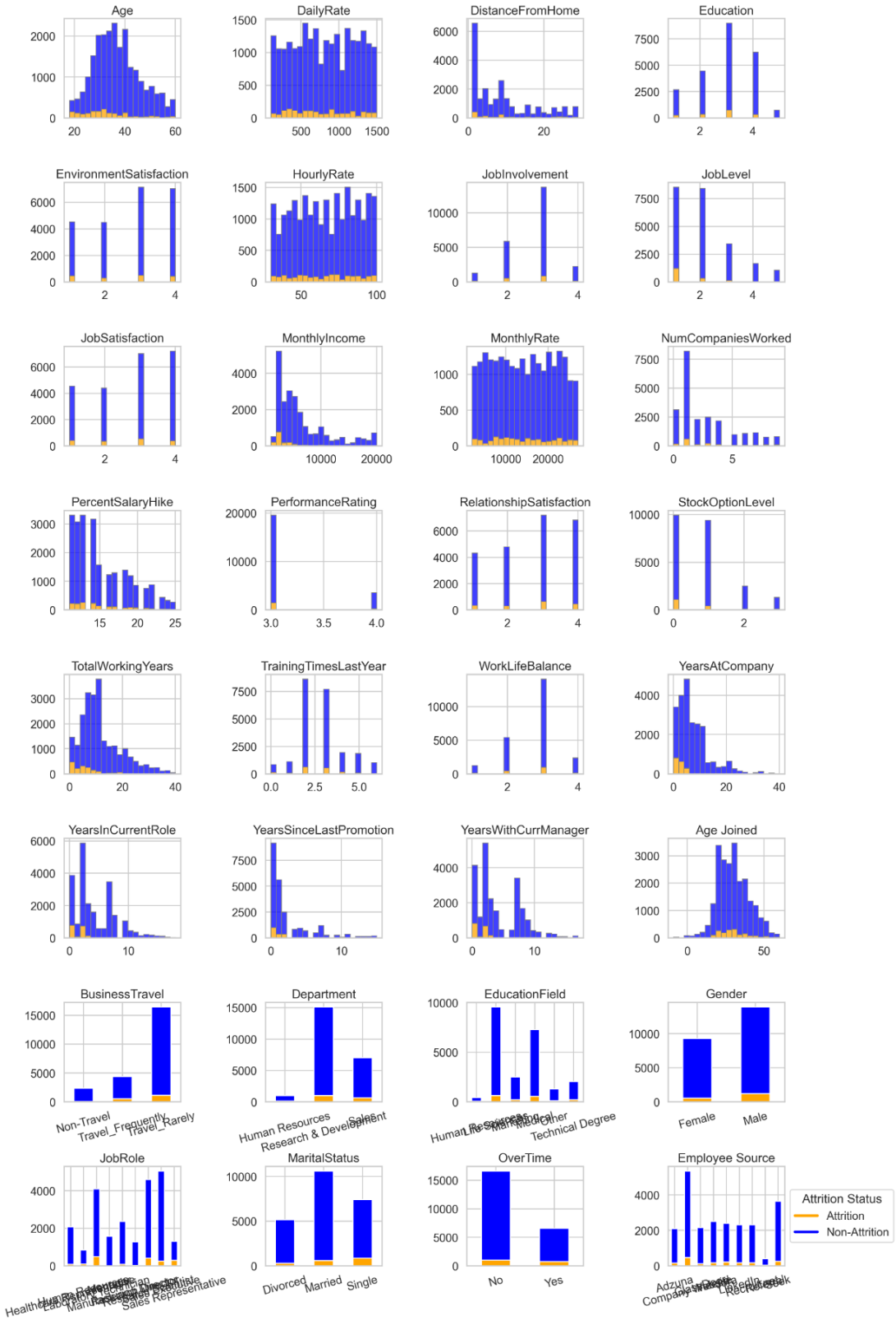


Figure 11: Relationship between all features and employee attrition

Figure 11 illustrates the distribution of Employee Attrition across a range of features within the dataset. A pronounced class imbalance is evident, as depicted in Figure 6, with a significantly higher occurrence of Non-Attrition relative to Attrition events. This imbalance is indicative of the real-world scenario where employee attrition rates are generally low.

An observation of particular interest is the distribution across gender and overtime variables. A gender imbalance is present, with male employees outnumbering females. This disparity must be considered during model interpretation, as it may impact the predictive performance and insights drawn from the model. The OverTime variable may also require careful consideration as there are significantly more "No" samples than "Yes" samples.

In analyzing employee demographics, factors such as Age, Education, and Job Involvement are normally distributed. Conversely, job-related factors like Monthly Income, Percent Salary Hike, Job Level, and Years at the Company exhibit a right-skewed distribution, indicating a higher concentration of employees at the lower end of the spectrum for these variables.

The exploration of feature distributions reveals pivotal insights for predicting Employee Attrition. Some demographic attributes such as Age, Education, and Job Involvement align with a normal distribution, thus fitting well with models that operate under the normality assumption. In contrast, some job-related factors like Monthly Income, Percent Salary Hike, Job Level, and Years at the Company are notably right-skewed. This skewness suggests a concentration of employees with lower values in these measures. Given these skewed distributions, models that assume a normal distribution for all features, such as Gaussian Naive Bayes, might not perform optimally. In contrast, tree-based models like Decision Tree and XGBoost are inherently more accommodating of features with skewed distributions due

to their non-parametric nature. These considerations will be taken into account in the predictive modelling methodology section.

III. Correlation analysis and multicollinearity

Multi-collinearity occurs when two or more independent features in a model are highly correlated. This can cause the standard error of the model's coefficients to increase (Daoud, 2017), which can cause the model to be biased and imprecise in its estimation of the significance of each predictor. (Yoo et al, 2017). Furthermore, the correlated variables can overshadow each other's effects, making interpretation of feature importance challenging and potentially leading to misguided conclusions. As such, the presence of highly correlated features can compromise the stability and predictive accuracy of machine learning models.

A visual inspection of variables in the dataset reveals the potential for high correlation between some features. For example, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, and YearsWithCurrManager all refer to the time an employee has spent working in the company, either overall or within specific roles or managers. A contextual understanding suggests that these columns are very likely to be highly correlated.

We use Pearson's correlation metric to measure collinearity between features.

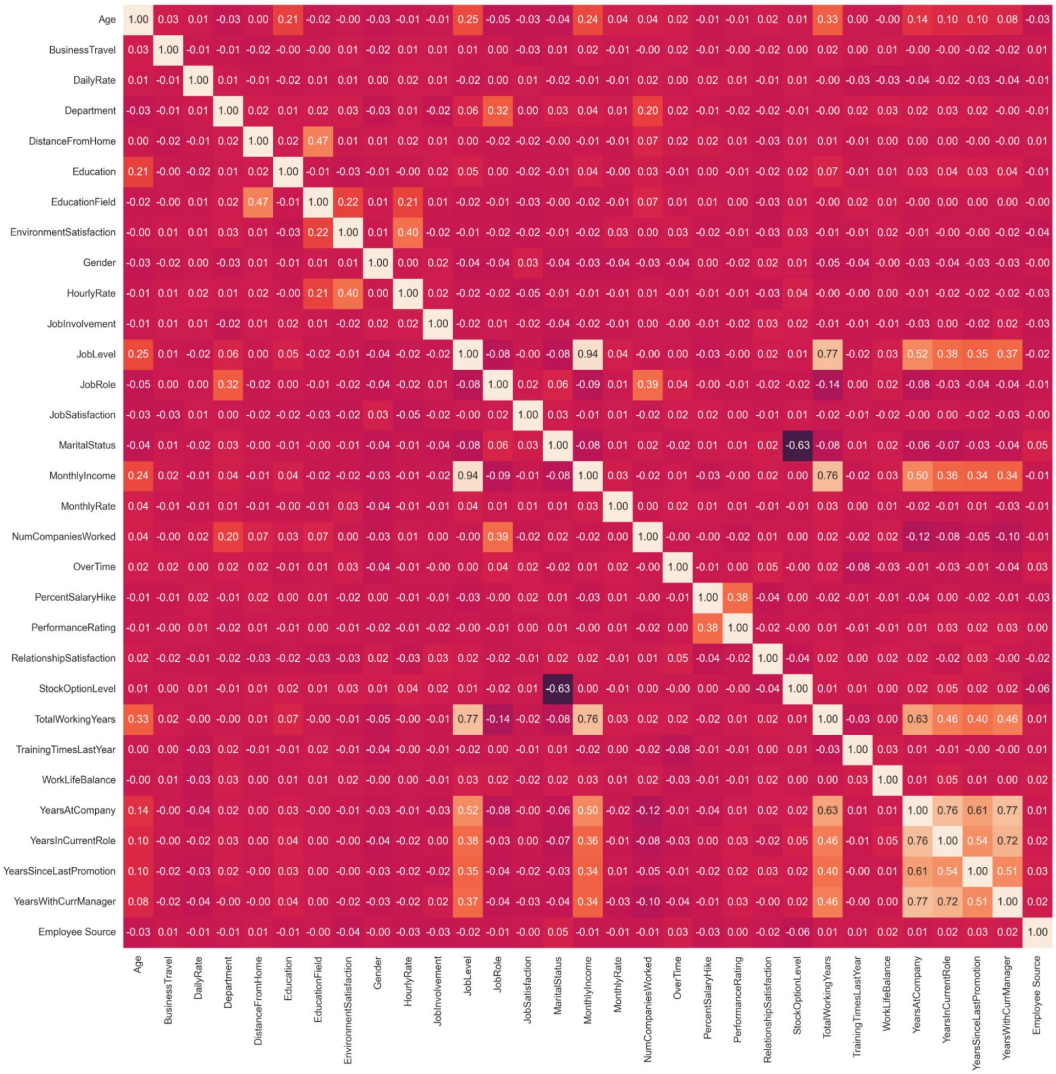


Figure 12: Correlation matrix between predictors

In the correlation matrix, we observe high correlation between YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager. Given that YearsAtCompany is used to split the target variable Attrition into "Attrition" and "Non-Attrition" values, including YearsAtCompany or other highly correlated columns could cause data leakage and result in biased model performance and explainability results. As such, we will remove these features.

Further, there is significant correlation among JobLevel, MonthlyIncome, and TotalWorkingYears, with JobLevel and MonthlyIncome having a high correlation coefficient of 0.94. Such a high correlation coefficient suggests that the variability in MonthlyIncome can be largely predicted by changes in JobLevel, so we will examine their relationship in a violin plot.

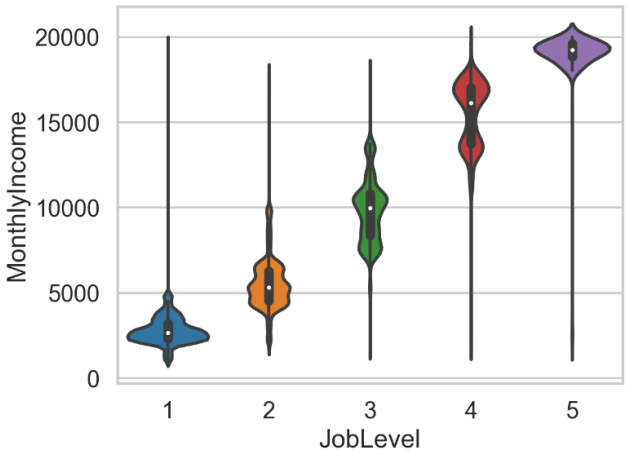


Figure 13: Violin plot of Monthly income by Job level

Clearly, job level and monthly income exhibit a distinct relationship. Different levels of the job correspond to distinct monthly incomes with minimal overlap. Given the utility of the MonthlyIncome feature in this context, we will retain it while removing JobLevel.

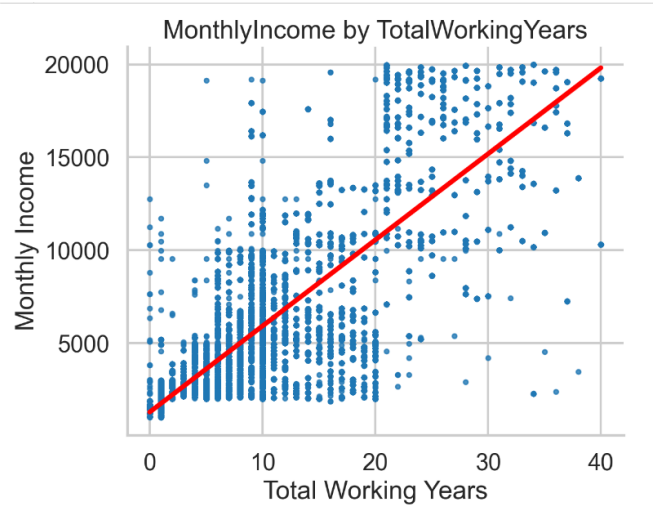


Figure 14: Scatter plot of Monthly income by Total working years

While "Monthly Income" and "Total Working Years" exhibit a strong correlation of 0.77 and are clearly correlated in the scatter plot, they represent distinct facets of an employee's professional profile, making it essential to retain both for a comprehensive analysis.

Chapter 4: Predictive Modelling Methodology

Model selection

Since the target variable of Attrition is a binary categorical variable with "Attrition" and "Non-Attrition", binary classification models will be used for the proposed modelling.

The models commonly used by Zhao et al. (2019) and Falluchi et al. (2020) include Logistic regression, Gaussian Naive Bayes, Support Vector Machine, Decision Tree Classifier, Random Forest Classifier and Extreme Gradient Boosting (XGBoost Classifier)

Listed below are the advantages and disadvantages of these models:

Model	Advantages	Disadvantages
Logistic Regression	• Simple and fast to train • Easily interpretable. • Less likely to overfit with small datasets	• Assumes linearity between variables and log-odds of outcome. • Not suited for complex relationships. • Performs poorly with missing data.
Gaussian Naive Bayes	• Simple and fast. • Works well with categorical data. • Less likely to overfit with small datasets	• Requires features to be normally distributed • Can perform poorly if independence assumption is violated.
Support Vector Machine	• Effective in high-dimensional spaces. • Good for non-linear data.	• Sensitive to noisy data and outliers. • Computationally intensive for large datasets.
Decision Tree Classifier	• Easily interpretable. • Requires little data preprocessing. • Can handle both numerical and categorical data. • Can handle missing data well.	• Prone to overfitting, especially with small datasets or complex trees. • Can be unstable with small variations in data. • Biased with unbalanced classes.

Random Forest Classifier	• Can handle large datasets • Reduces overfitting compared to decision trees. • Can handle missing values well.	• Can be slow to predict with a large number of trees. • Not as interpretable as individual decision trees. • Training can be computationally intensive.
XGBoost Classifier	• Capable of capturing complex relationships in data. • Provides feature importance, aiding in interpretability. • Uses gradient boosting, a machine learning framework known for high performance	• Can be prone to overfitting if not tuned correctly. • Interpretability is not as clear as simpler models; requires feature importance techniques like SHAP • Requires careful hyper-parameter tuning to maximize performance

Table 1: Advantages and disadvantages of selected classification models

Logistic regression has been used in previous studies on attrition, and has been praised for its simplicity and interpretability. (Harsha et al., 2020; Fallucchi et al., 2020). Chen (2023) further highlighted the interpretability of logistic regression by assessing the statistical significance of each predictor to determine the key factors impacting attrition. However, they also recognised its limited accuracy and weaknesses in handling the attrition problem, compared to other models.

XGBoost is a gradient boosting model that has gained popularity for its speed and accuracy in recent years. It is highly accurate and has built-in features for missing data handling, cross-validation, regularization, and parallel processing.(Kakad et al., 2020). Zhao et al. (2019)

found XGBoost to be the best performing model in his comparison of ten different classifiers on HR datasets of varying sizes. Mohiuddin et al. (2023) reached a similar conclusion when comparing eight different classifiers.

In the scope of this study, we will focus on two models - Logistic regression and XGBoost. These models were selected for their efficacy in prior research. Logistic Regression serves as our baseline model, chosen for its clear interpretability compared to other models. XGBoost will be utilised as the final model for its very high accuracy and predictive capability compared to other models.

Objectives

The primary objective of the predictive classifier in this study is to accurately determine if an employee is likely to attrit based on employee demographic or job-related factors. This enables companies to proactively identify employees at risk of leaving, thus facilitating timely intervention measures. Besides, this model will help to understand the extent to which demographic and job-related factors influence attrition. This will highlight the main causes of attrition and shape better-informed HR strategies and policies.

Data preprocessing

Shuffling the dataset. The dataset was shuffled before splitting. This is important as the employees were listed sequentially based on their employee number. Shuffling the data ensures both the training and testing set is uniformly distributed and are representative of the overall dataset.

Handling imbalanced classes. SMOTE is used to balance the classes, due to the significant class imbalance in the dataset. This method generates synthetic samples from the minority class, which helps to prevent the model from being biased towards the majority class. By achieving a more balanced class distribution, the model's ability to identify and predict minority class instances (in this case, employees likely to attrit) is enhanced.

Train-test split. Since there is only one dataset present, it is split into a training set and testing set. We have chosen a 70-30 train-test split, where 70% of the records go into the training set and 30% into the testing set. The training set contains 16 231 rows and is used for training the machine learning model. The testing set contains 6 957 rows and is used to test the performance of the machine learning model.

Model evaluation techniques

Stratified sampling. We employed stratified sampling during the data splitting process. This ensures that both training and test sets maintain consistent proportions of each class, mirroring the overall distribution of attriters and non-attriters in the dataset. Given the significant class imbalance in our dataset, this approach is vital. It prevents biased training or testing outcomes by accurately representing both classes in the training and testing stages.

Cross-validation with holdout. Cross-validation with holdout involves initially partitioning the dataset into a training set and a separate test (or holdout) set. The training set then undergoes k-fold cross-validation, where it's divided into k subsets (or folds). During each of the *k* iterations, the model trains on *k-1* of these folds and validates on the remaining fold. This ensures that every fold gets the chance to serve as the validation set once. The final performance metric is derived by averaging the outcomes from each iteration. K-fold cross

validation offers a more comprehensive and thorough evaluation of the model's ability than train-test split as it uses the entire dataset for training and testing. Subsequently, the model is assessed using the holdout test set to gauge its performance on entirely unseen data. Fallucchi et al. (2020) highlighted similar advantages of this approach, illustrating it in their study with Figure 15.

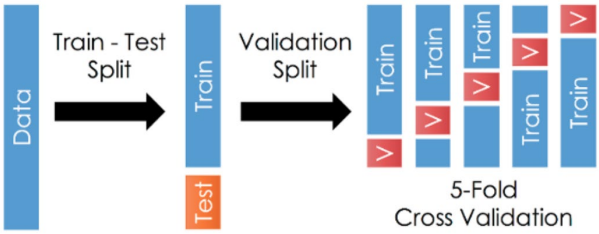


Figure 15: K-Fold cross validation with holdout test set,

Hyper-parameter tuning

Hyper-parameter tuning refers to systematically searching for the optimal set of hyper-parameters (parameters not directly learned from the data) to achieve the best performance in machine learning models. Zhao et al. (2019) demonstrated this using Grid Search technique which tests every combination of hyper-parameters defined in a specific range (Pedregosa et al., 2011). However, researchers have found Random Search to be more "practical" than Grid Search, citing its efficiency in high-dimensional spaces (Bergstra and Bengio, 2012) and better performance with small numbers of hyper-parameters. (Liashchynskyi and Liashchynskyi, 2019).

As such, Random Search will be utilised for hyper-parameter tuning of both models. Information about the exact hyper-parameter ranges used can be found in Appendix B. Details on the Python environment, code and code output can be found in Appendix E and F.

Model performance metrics

A confusion matrix is used to evaluate the performance of a classification model. It compares the actual target classes with those predicted by the machine learning model. In a binary classification problem as our problem statement, the confusion matrix will be as follows:

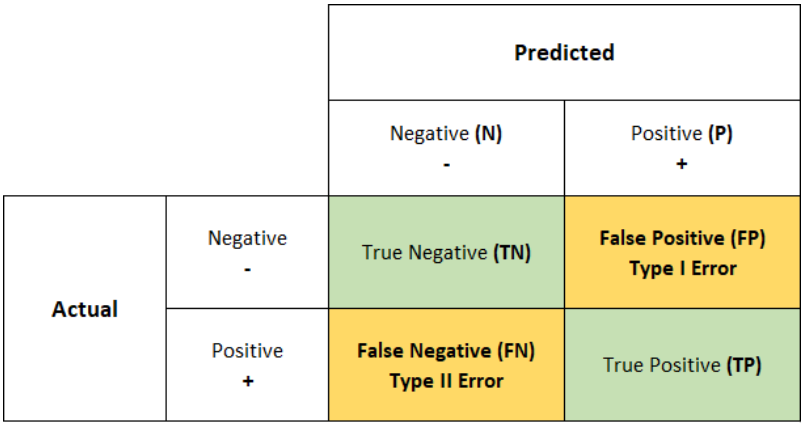


Figure 16: Confusion matrix (Suresh, 2020)

The possibilities include:

- True Positive (TP): These are cases in which the model predicted positive, and it's true.
- True Negative (TN): The model predicted negative, and it's true.
- False Positive (FP): The model predicted positive, but it's false and the actual class is negative. (Type I Error)
- False Negative (FN): The model predicted negative, but it's false and the actual class is positive. (Type II Error)

In the context of employee attrition, the errors in prediction would result in:

- False Negative (FN): Predicting an employee will not leave, but they actually leave.
- False Positive (FP): Predicting an employee will attrit, but they actually stay.

Recall. In most companies, the costs of training and retaining new staff is significant. (Navarra, 2022). Predicting that an employee will stay will result in much costs and effort being spent in training them. Erroneously predicting an employee who leaves as someone who will stay can lead to wasted resources invested in their training and development. (Mello, 2011). Given these stakes, it is crucial to minimize false negatives and prioritize maximizing recall. This aligns with Falluchi et al. (2020), who also used recall as the primary performance metric in their comparisons.

ROC-AUC. To measure overall model performance on an imbalanced dataset, accuracy can be misleading as a model might achieve high scores by simply predicting the majority class. In contrast, ROC-AUC evaluates a model's ability to differentiate between classes across varying thresholds, making it a more reliable metric. Zhao et al. (2019) also adopted ROC-AUC as the primary performance metric in their comparison of classifiers on an attrition dataset.

Chapter 5: Model Evaluation and Results

Using SMOTE and hyper-parameter tuning, these are the summarised model evaluation results based on the performance metrics detailed above. All performance metrics quoted are from the holdout testing set. The set of optimal hyper-parameters obtained through random search will be listed in Appendix C.

Model	**ROC-AUC**	Precision	**Recall**	Accuracy
Logistic Regression	**0.8267**	0.2790	**0.5766**	0.8564
XGBoost Classifier	**0.9987**	1.0	**0.9943**	0.9996

Table 2: Summarised model performance metrics of logistic regression and XGBoost Classifier

Confusion Matrix (**Logistic Regression**)		Predicted		Percentage Correct
		Non-Attrition	Attrition	
Actual	Non-Attrition	5657	778	87.91%
	Attrition	221	301	57.66%
Overall Percentage Correct				85.64%

Table 3: Confusion matrix of logistic regression model

Confusion Matrix (**XGBoost Classifier**)		Predicted		Percentage Correct
		Non-Attrition	Attrition	
Actual	Non-Attrition	6435	0	100.00%
	Attrition	3	519	99.43%
Overall Percentage Correct				99.96%

Table 4: Confusion matrix of XGBoost Classifier model

Based on the model results from Table 2, XGBoost Classifier greatly outperforms Logistic Regression in both ROC-AUC and recall metrics:

- XGBoost achieves a very high ROC-AUC of 0.9987 compared to Logistic Regression at 0.8267.
- XGBoost achieves a very high Recall of 0.9943 compared to Logistic Regression at 0.5766. This indicates that of all the employees that attrit, XGBoost can correctly identify 99.43% of them, compared to just 57.66% for Logistic Regression.

XGBoost's Recall score of 0.9943 means that it is highly effective in predicting potential attriters, minimizing the chances of overlooking employees at risk of leaving. In contrast, a

Recall of 0.5766 for Logistic Regression suggests that just under half of the potential leavers might go unnoticed, thereby hindering proactive talent management efforts.

Given XGBoost's impressive Recall score, it stands out as the more proficient model for identifying employees who are likely to leave, ensuring a more effective intervention strategy. In light of this, we will concentrate our analysis on the XGBoost model going forward, as it demonstrates superior performance in distinguishing between those who will stay and those who will depart.

One of the reasons for such a stark difference in performance between XGBoost and Logistic Regression could be the relationship between the features and the target variable being nonlinear and complex. While Logistic Regression assumes a linear relationship between the predictors and the log odds of the target, XGBoost, being a gradient boosting algorithm, can capture more intricate patterns in the data by building sequential trees. (Chen and Guestrin, 2016). This might explain why XGBoost vastly outperforms Logistic Regression in both ROC-AUC and Recall metrics for this particular dataset.

While the XGBoost Classifier's very high performance on the dataset is promising, we remain cognizant of the potential risk of model overfitting, where the model is over-reliant on the training data and generalizes poorly to new data. To validate our model's robustness, we systematically examined the following:

- Cross-Validation Consistency: Model performance was evaluated across multiple folds to ascertain its stability across different training data.
- Hyperparameter Analysis: The model's performance was evaluated over a range of hyperparameter settings to understand its behavior under different configurations.

- Feature Importance Review: An analysis was conducted to study the model's reliance on specific features. This can be found in the feature importance section later in this report.

Cross-validation consistency

Utilizing k-fold cross-validation, we examined the model's consistency by comparing its performance across all folds. Specifically, we assessed the mean and standard deviation of ROC-AUC scores over the 5 folds. A close clustering of scores with a small standard deviation would indicate consistent performance across different data subsets, underscoring the model's reliability.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	**Mean ROC-AUC**	**Standard deviation**
XGBoost Classifier	0.9992	0.9992	0.9994	0.9949	0.9994	**0.9984**	**0.001762**

Table 5: ROC-AUC scores across cross-validation folds: XGBoost Classifier

As shown in Table 5, XGBoost Classifier exhibits consistent ROC-AUC scores across cross-validation folds, as evidenced by the low standard deviation. This indicates that the XGBoost model demonstrates high levels of stability and is not overly sensitive to the specific characteristics of each CV fold. Such consistency suggests that the model is likely to generalize well to other subsets of the dataset, reducing the risk of overfitting.

Hyper-parameter analysis

We conducted a comprehensive hyperparameter tuning using RandomSearchCV, assessing the model's performance over a wide range of hyper-parameters. The evaluation aims to

determine the optimal hyperparameters while examining the stability of the model across a wide range of different hyper-parameters. A model that maintains a high level of performance across a range of hyperparameter values is less likely to be overfitted to the training data and is more adaptable to new data, reinforcing confidence in its reliability.

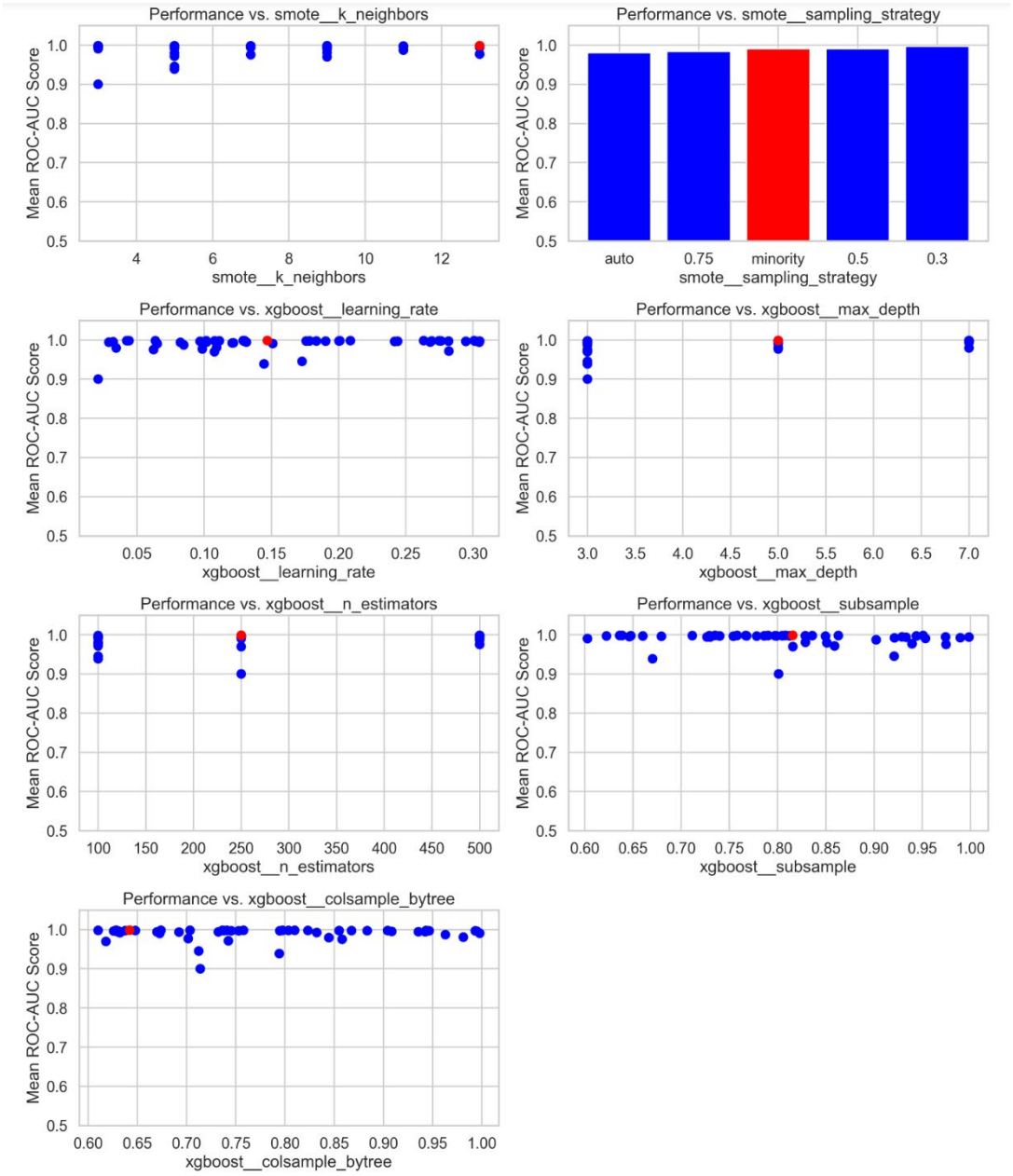


Figure 17: XGBoost Classifier's mean ROC-AUC across folds, across 50 iterations using different hyper-parameters. Red dot highlights the optimal model's mean ROC-AUC

Figure 17 illustrates the XGBoost Classifier's ROC-AUC score across 50 distinct combinations of SMOTE and XGBoost hyper-parameters. Notably, the largely similar performance (a few outliers notwithstanding) across different values of xgboost__subsample and xgboost__colsample_bytree suggests that the model is robust to the specific selection of dataset samples and columns in the training data.

This consistent performance across a diverse range of hyper-parameter configurations underscores XGBoost's capacity to accommodate different scenarios, achieve a balanced model complexity, and effectively learn patterns from its features, thereby minimizing the risk of overfitting.

Chapter 6: Model interpretation

As highlighted in the Literature Review section, this section will focus on the interpretation of the XGBoost Classifier model using SHAP.

Overall feature importance and influence of individual features on attrition

The SHAP summary plot is used to visualize feature importances, revealing the impact and directionality of each feature on the model's predictions. Features are arranged in descending order of feature importance. The SHAP summary plot combines Shapley values on the x-axis with features on the y-axis. (Mohiuddin et al., 2023). The feature values are colour-coded from low (blue) to high (red). For example, with the TotalWorkingYears feature, the blue points indicate employees with few working years, while red points indicate employees with more working years. In this case, employees with few working years tend towards attrition while employees with more working years tend towards retention.

In addition, SHAP dependence plots contain a more granular visualization of the relationship between feature values and model predictions. They can showcase non-linear relationships and unique clusters or trends, providing a more detailed analysis. Interaction effects are visualized by color-coding the data points based on the values of another feature, highlighting potential relationships and dependencies between two features in their influence on the model's prediction.

Further, the SHAP decision plot provides a detailed view of the prediction path for individual samples, distinguishing between attriters and non-attriters. Each line represents a single data point, illustrating the cumulative effect of each feature on the model's output. By overlaying 50 samples, the plot aggregates how features influence predictions across varied instances, offering insights into both global feature importance and individual prediction determinants.

It is especially helpful to consider these plots in conjunction with one another when analysing the impact of individual features. Previous research, notably in a medical research paper by Bloch and Friedrich (2021), has utilised the SHAP summary plot to gain insights into the individual contributions of features to the prediction.

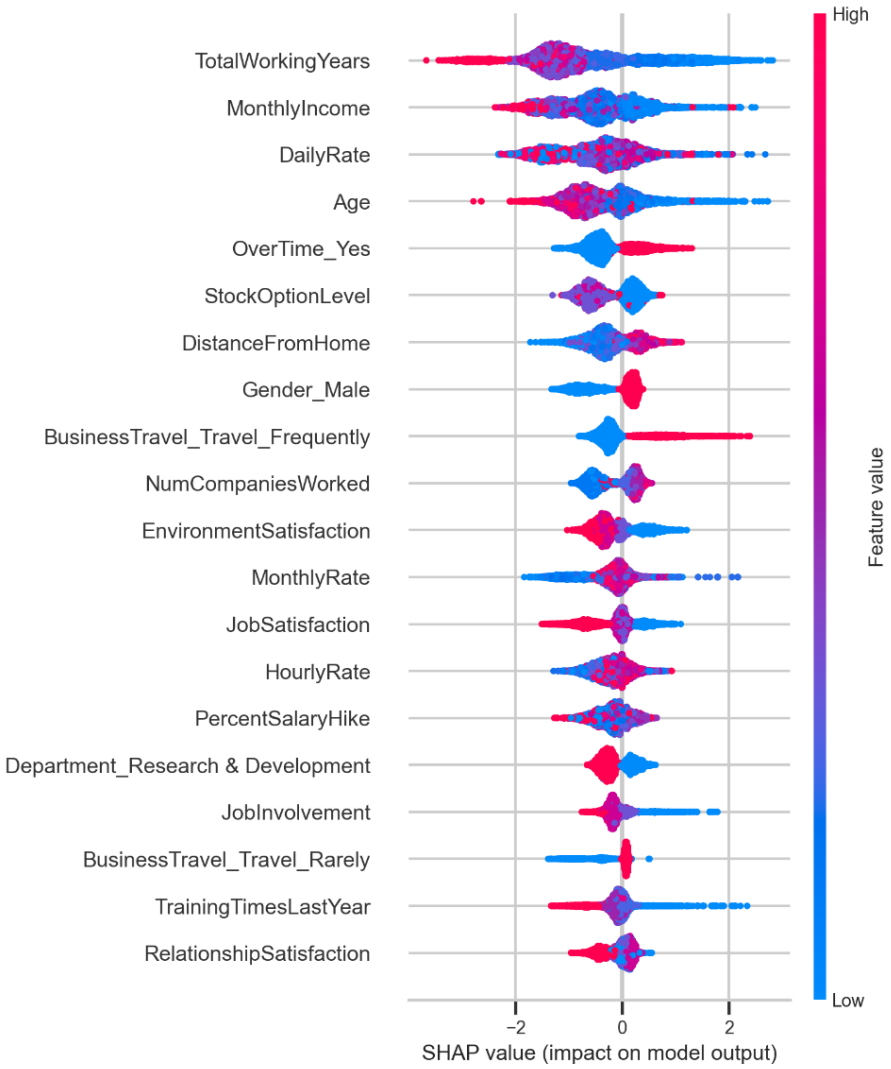


Figure 18: SHAP summary plot

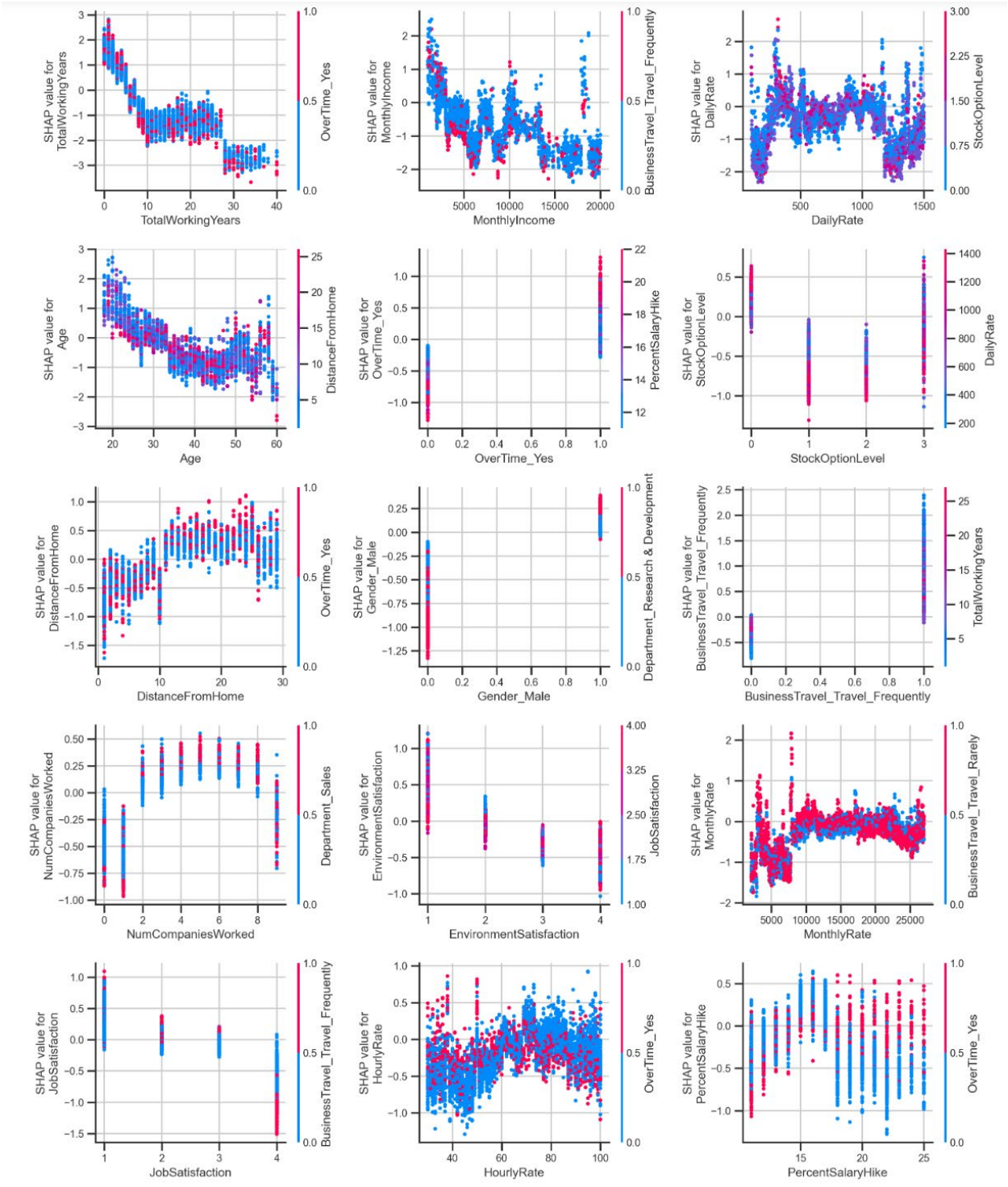


Figure 19: SHAP dependence plots

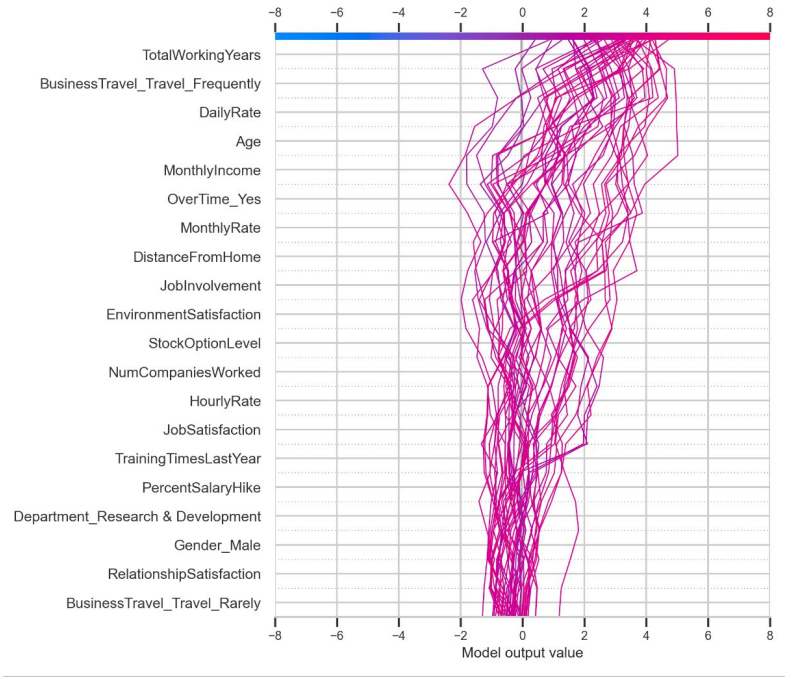


Figure 20: SHAP decision plot for attriters

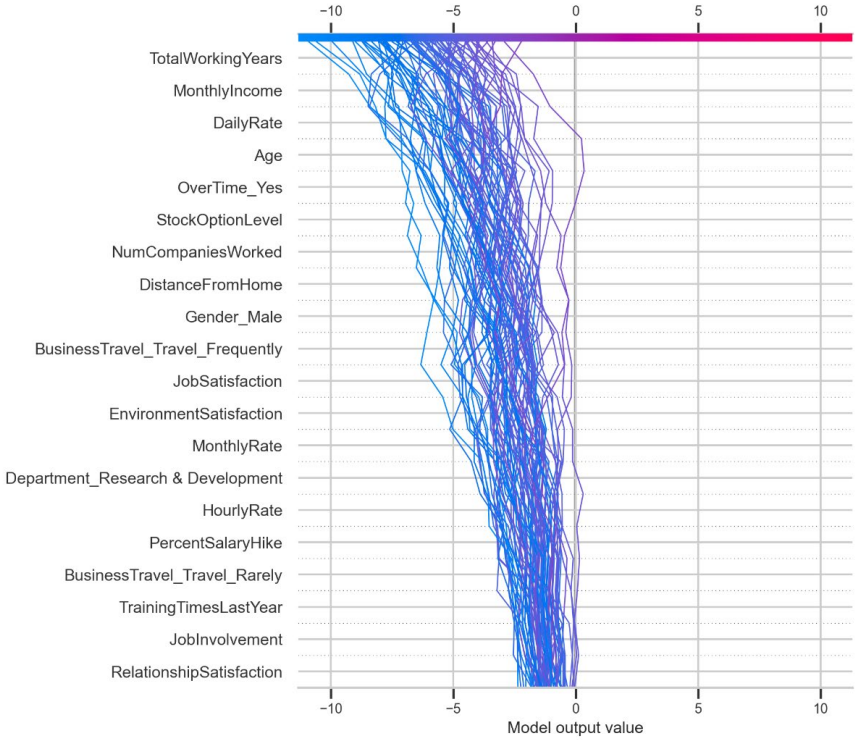


Figure 21: SHAP decision plot for non-attriters

The summary plot indicates that, in descending order of SHAP feature importance, work experience, income, age, overtime work, stock options, distance from home, gender, business

travel frequency, number of companies worked, and job satisfaction have a sizeable impact on attrition. For clarity, we've categorized some of these influential features into two primary groups: "Employee demographics," representing personal attributes of the employee, and "Job-Related factors," denoting characteristics associated with their professional role.

Employee demographics:

- **TotalWorkingYears:** A key predictor influencing attrition, with the highest feature importance. The dependence plot indicates an inverse relationship, highlighting that employees with more work experience have a lower probability of attrition. It further indicates that the probability of attrition decreases sharply from 0-10 years, stagnates around 10-28 years, and then reaches the lowest point beyond 28 years. This is in line with the common knowledge that employees tend to switch jobs often early on in their career. In the context of this dataset, employees with less than 10 years' work experience have a higher tendency to leave more quickly.
- **Age:** A significant predictor that influences attrition. The dependence plot indicates an inverse relationship, highlighting that older employees generally have a lower probability of attrition. It is also observed that employees under the age of 30 have a far higher concentration of positive SHAP values than other age groups, indicating that employees younger than 30 have a far higher propensity of attriting. The interaction with Distance from Home suggests that for employees 30 years or older, greater Distance from Home appears to be related to higher probability of attrition. (higher concentration of red points near the upper range of SHAP values at age > 30 range)
- **Gender_Male:** Unlikely to play a large role in influencing attrition. The spread of the values in the summary plot and presence of a wide range of SHAP values spread

across both genders in the Gender_Male dependence plot indicates variability in its influence. The decision plot also does not show a clear pattern favouring either gender, with it not having a significant impact on attriters and non-attriters alike. This suggests that the impact of gender on attrition may be diminished or influenced by interaction effects, such as distribution of genders within departments.

Job-related factors:

- **MonthlyIncome:** A key predictor influencing attrition, with the second highest feature importance. The dependence plot depicts an inverse relationship, where employees with higher monthly income are less likely to attrit. This is more obvious for incomes below $10,000, where there's a higher concentration of positive SHAP values suggesting a higher possibility of attrition. The decision plots further underscore the impact of monthly income on attrition decisions. For non-attriters, a higher monthly income significantly contributes to their retention. In contrast, for attriters, while monthly income plays a role, its influence on attrition is comparatively smaller.
- **DailyRate:** An influential predictor for attrition as reflected in Figure 18, ranking third in terms of feature importance. It has a complex relationship with huge fluctuations in SHAP values, showing an unreliable impact on the model's prediction. The interaction with Stock Option Level indicates that employee attrition is likely to also be influenced by stock option provisions, making the impact of Daily Rate harder to quantify. The decision plots reveal that for non-attriters, the impact of the Daily Rate is diverse: in some cases, it strongly influences retention, while in others it has a minimal or opposing effect. In contrast, for attriters, the Daily Rate consistently contributes to their decision to leave. This suggests that the Daily Rate has a more

uniform, positive influence on attriters, whereas its effect varies more significantly among non-attriters.

- **OverTime_Yes:** A significant predictor that influences attrition. Both the summary plot and dependence plot show that employees who work overtime typically exhibit positive SHAP values, indicating a higher likelihood of attrition. Conversely, employees who do not work overtime have a lower risk of attrition. The interaction effects suggest that the impact of overtime work is also contingent on an individual's work experience, their commute and their recent compensation adjustments. In other words, overtime work does contribute to employee attrition though its impact is varied. Unsurprisingly, employees who do not work overtime and receive salary hikes have the lowest attrition propensity.
- **StockOptionLevel:** A predictor that influences attrition. The dependence plot shows a spread across a large range of SHAP values for each stock option level. The SHAP values suggest that employees with lower stock option levels are more likely to attrit, while those with higher levels are less prone to attrition. The decision plots further show that while stock options have a bigger impact on non-attriters than attriters, the impact is quite varied across both, making the pattern less discernible. The wide range of interaction effects from Daily Rate suggest other factors may be influencing the overall attrition propensity.
- **BusinessTravel_Travel_Frequently:** A predictor that influences attrition. Both the summary plot and dependence plots show a wide spread of positive SHAP values and narrow spread of negative SHAP values near zero. Further, the decision plot shows that frequent business travel has a huge positive influence on attriters, while it has little influence on non-attriters. This indicates that while frequent business travel can increase employees' attrition propensity, not having frequent business travel also

doesn't boost retention. It is likely that frequent business travels isn't well-received by some employees while others are just not affected by it, so its impact can be varied.

- **Environment & Job Satisfaction:** Predictors that influence attrition. These two features are closely intertwined due to their high interaction effects and similar domain. Both satisfaction columns exhibit an inverse relationship with attrition, so higher satisfaction is linked to a lower attrition propensity. While this correlation is evident across summary, decision and dependence plots, it is clear from the feature importance that satisfaction isn't the largest influential factor in determining attrition. Employees weigh other factors more heavily in their decision-making regarding continued employment.

Chapter 7: Recommendations

This chapter builds upon the comprehensive insights gained from the explainability analysis conducted in Chapter 6, where SHAP (SHapley Additive exPlanations) plots played a pivotal role in interpreting the predictive model's outcomes. Drawing on these findings, we propose targeted recommendations aimed at addressing the factors influencing employee attrition at Winner Engineering.

Career Development for Younger and Less Experienced Employees:

The SHAP analysis reveals that the probability of attrition declines with increased work experience and age, reaching a plateau after around 10 years of service and beyond the age of 30. Tomaskovic-Devey and Orellana (2022) emphasize the significance of offering transparent career advancement paths to younger employees to foster a perception of a long-term, stable relationship with the company.

In line with these insights, these programs can be implemented:

- Initiation of a structured mentorship program, matching junior staff with experienced mentors, to facilitate knowledge transfer and professional guidance.
- Development of clear career pathways, complemented by developmental milestones, aimed at bolstering retention and stimulating the ambition of younger employees. This should incorporate annual performance evaluations to encourage and recognize continuous improvement and excellence.
- Provision of financial support for skill-enhancement courses, enabling younger employees to acquire new competencies in alignment with their personal career aspirations.

These strategies aim to establish a nurturing environment that not only acknowledges but actively invests in the professional growth of younger staff, increasing employee retention.

Employment Benefits for More Experienced Employees:

While the attrition rates are comparatively lower for seasoned employees, maintaining their engagement and satisfaction is essential to promote retention. Additionally, the implementation of hybrid work models would particularly benefit staff over the age of 30 who may be exhausted from long commutes, as shown by the interaction effect of DistanceFromHome on Age.

In line with these insights, we recommend the following strategies:

- Enhancement of employee benefits to recognize the seniority and contributions of experienced staff, including more generous leave policies and adaptable work

arrangements. This approach not only rewards tenure but can also inspire junior employees to envisage a long-term future with the company.

- Introduction of hybrid work arrangements to support a balance between personal and professional responsibilities, especially benefiting those negatively impacted by long commutes. This demonstrates the company's commitment to flexibility and the importance placed on employee well-being.
- Establishment of annual recognition and performance bonuses for employees who have demonstrated significant contribution. This serves as a motivation for continued excellence and conveys the company's appreciation for their dedication and hard work.

These strategies aim to cultivate an environment that values long-standing employees and provides tangible benefits, thereby fostering loyalty and encouraging long-term retention.

Recommendations for Compensation Management:

The SHAP analysis underscores that salary metrics, notably monthly income and daily rate, are pivotal in determining employee attrition, highlighting a trend where lower-paid employees exhibit a higher likelihood of turnover. The importance of pay transparency cannot be overstressed in this context. Lam et al. (2022) highlighted that transparent compensation practices foster trust, fairness, and job satisfaction among employees, factors that collectively have a "significant positive effect" on employee retention. (Xuecheng et al., 2022).

In line with these insights, we recommend the following strategies:

- Introducing loyalty bonuses at key tenure milestones can serve as a significant retention tool. Specifically, providing bonuses at 12 months and again at 36 months aligns with the 25th and 50th percentiles of current employee tenure. This strategy not only rewards tenure but also serves as a tangible acknowledgment of employee contributions over time, which can increase feelings of appreciation and job satisfaction.
- Transition to a more transparent pay structure to clearly communicate compensation determinants. This can include the disclosure of estimated salary ranges for different roles and the criteria for salary increases. Reassuring employees that they are paid fairly and equitably can increase employee retention.

Collectively, these strategic measures are designed to create a workplace culture that tangibly rewards career longevity through the use of financial benefits, aligning with the importance that most employees place on compensation-related factors.

Promoting Employee Work-Life Balance:

The SHAP analysis shows that overtime work and frequent business travel have a largely positive influence on attrition. Ambrose et al. (2021) demonstrated a relationship between frequent business travel and an increase in employees' intentions to leave, while Shao (2022) associated extended work hours with reduced job satisfaction, subsequently affecting attrition rates. These insights accentuate the critical role of work-life balance in retaining employees, emphasizing the necessity of ensuring that professional responsibilities do not detract from the quality of their personal lives (Kabir & Tirno, 2018).

In line with these insights, we recommend the following strategies:

- Adopt flexible work schedules by granting employees the flexibility to design their work hours and location, a change that has been shown to enhance job satisfaction (Wu & Zhou, 2020). This measure provides employees with the necessary autonomy to align their professional responsibilities with personal obligations, leading to a more satisfied and productive workforce.
- Provide employees task autonomy, giving them the control over the sequence and method of their tasks. This autonomy can alleviate the negative impacts of time pressure on job satisfaction, creating a more committed and engaged team. (Häusser et al., 2010).

Chapter 8: Limitations and Future Work

There are three key limitations in this study, which will be addressed. Further, we will discuss future endeavours to expand the analysis of this project.

Employment dynamics in the post-pandemic era

The primary dataset utilized was originally released by IBM in 2017, which predates significant changes in the employment market due to the COVID-19 pandemic. A recent study by Basiouny (2022) underscores the emergence of new employee expectations post-pandemic, highlighting an increased demand for flexibility, competitive remuneration, and enhanced work-life balance. Given that the dataset precedes these developments, it does not encapsulate the substantial shifts in recruitment, employment conditions, and organizational cultures that have since become prevalent, as documented by Ancillo et al. (2023). As such, the current dataset may not accurately reflect the latest trends in hiring practices, employee expectations, and organizational culture, potentially limiting the predictive accuracy of the derived models for post-pandemic employment conditions.

Generalizability of IBM HR data

The dataset was sourced from IBM HR data. Since IBM is a large multinational technology company with estimated 290,000 employees worldwide (Fortune, n.d.), its employee distribution and hiring dynamics may differ from that of smaller firms such as Winner Engineering. This may pose a challenge to the generalizability of the model's findings.

Different proportion of class imbalance

Based on observations from the dataset, IBM's attrition rate is notably low at approximately 8.12%. This contrasts with companies like Winner Engineering, which exhibit a significantly higher attrition rate. This results in the proportion of class imbalance differing greatly. As such, SMOTE hyper-parameters for the model trained on IBM dataset may not be optimal when applied to data from Winner Engineering.

Future research endeavours

To address this limitation, future research endeavours will aim to source more recent HR datasets, preferably containing data within the last 1-2 years. This ensures that the data better reflects the changes in modern workspace as detailed by Ancillo et al. (2023), thereby providing a more accurate and relevant foundation for exploratory analysis and predictive modelling.

Further, subsequent investigations will extend to evaluate the robustness of our model when applied to data from different organizations, particularly focusing on how the model performs across datasets with different attrition rates. Such assessment will be crucial in improving the scalability and applicability of the model to real-world HR data.

Chapter 9: Deployment

The deployment of the model will be done in the HR Decision Support System developed in Tableau. Tableau is a powerful and versatile data visualization tool that enables users to create interactive and shareable dashboards. (Tableau, n.d.) It is popular in both business and technology fields for its simplicity and non-reliance on programming skills. This makes Tableau a good choice for rolling out a dashboard that aids HR decision-making and provides comprehensive reporting to upper management.

The HR Decision Support System is fully interactive and has two major sections. More information on how to set up and use the dashboard is found in Appendix D.

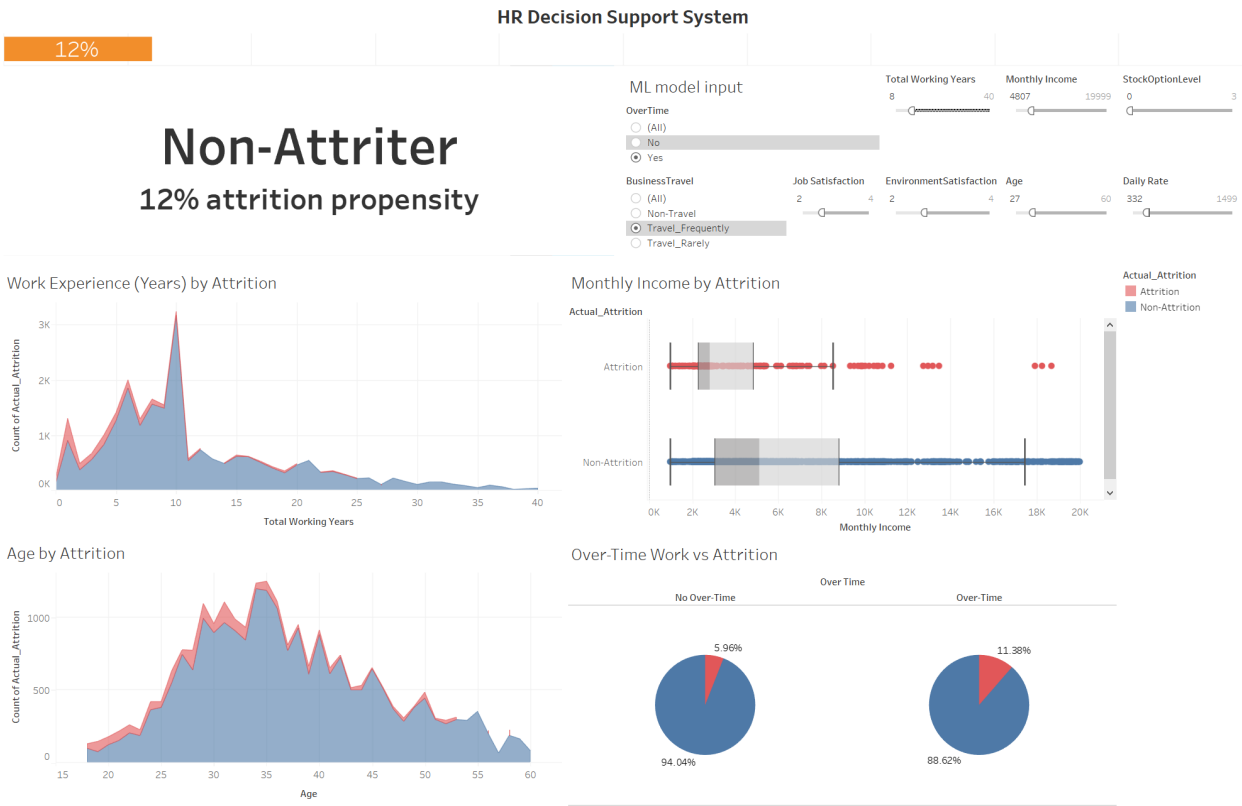


Figure 22: HR Decision Support System in Tableau. Full dashboard functionality available with TabPy client and an active connection to a running Jupyter Notebook instance.

The first section at the top of the HR Decision Support System includes the key employee demographic and job-related factors highlighted in the Model Interpretation section. End-users can enter the values for the employee using the sliders and buttons under "ML model input". Each time any of the values is changed, the model's prediction is immediately displayed. This allows the end-user to explore for themselves how each feature affects attrition propensity.

The second section at the bottom half includes several exploratory graphs for the end-user to explore the dataset and better understand the IBM HR dataset used to train the model.

Chapter 10: Conclusion

In conclusion, employee attrition emerges as a critical concern for many organizations. This can be attributed to the substantial financial burdens it incurs and the prevalent lack of understanding among employers regarding its underlying causes. The predictive model addresses the business objective by providing an in-depth analysis of the principal factors influencing employee attrition, alongside data-informed recommendations aimed at enhancing talent management and employee retention strategies. Furthermore, the data mining objective was fulfilled through the formulation of an XGBoost classifier that correctly predicts both attriters and non-attriters with over 99% accuracy. The investigation encompassed an extensive review of relevant literature, providing insights on potential contributing factors and elucidating the predictive modelling and model interpretation techniques employed in previous studies.

The process of data understanding and preparation revealed the imbalanced class distribution of Employee Attrition, which led to the use of Synthetic Minority Over-sampling Technique (SMOTE) to balance the class distributions. Within the predictive modelling phase, the k-fold cross-validation method was employed to develop an accurate and robust predictive model. The XGBoost Classifier's performance was markedly superior to that of the baseline Logistic Regression model in terms of Recall and ROC-AUC scores, thus it was selected for subsequent analysis and eventual deployment.

Insights drawn from the comprehensive examination of SHAP summary, dependence, and decision plots have led to a set of recommendations to improve employee retention. Moreover, HR and senior management have the capability to delve into the data independently through the interactive Tableau dashboard, utilizing the in-depth SHAP plot analyses to craft and propose tailored retention strategies.

The study recognized three primary limitations: modern employment dynamics, generalizability of IBM HR data and difference in proportion of class imbalance. These limitations can be mitigated by collecting more recent HR data from a wider range of companies to improve the model's generalizability and performance.

9002 words

(Word count includes descriptions and headers)

References

Alduayj, S. S., & Rajpoot, K. (2018). Predicting Employee Attrition using Machine Learning. In 2018 International Conference on Innovations in Information Technology (IIT) (pp. 93–98). https://doi.org/10.1109/innovations.2018.8605976

Ambrose, S. C., Waguespack, B. P., & Rutherford, B. N. (2021). The negative effects of travel friction among road warrior salespeople. Industrial Marketing Management. https://doi.org/10.1016/j.indmarman.2021.06.004

Ancillo, A. D. L., Gavrila, S. G., & Núñez, M. T. D. V. (2023). Workplace change within the COVID-19 context: The new (next) normal. Technological Forecasting and Social Change, 194, 122673. https://doi.org/10.1016/j.techfore.2023.122673

Basiouny, A. (2022, August 2). Employee turnover costs more than you think. Wharton Knowledge. https://knowledge.wharton.upenn.edu/article/why-employee-turnover-costs-more-than-you-think/

Basiouny, A. (2022, July 19). Finding balance in a post-pandemic workplace. Wharton Knowledge. https://knowledge.wharton.upenn.edu/article/finding-balance-in-a-post-pandemic-workplace/

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13, 281-305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

Bloch, L., Friedrich, C. M., & for the Alzheimer's Disease Neuroimaging Initiative. (2021). Data analysis with Shapley values for automatic subject selection in Alzheimer's disease data sets using interpretable machine learning. Alzheimer's Research & Therapy, 13, 155. https://doi.org/10.1186/s13195-021-00879-4

Boushey, H., & Glynn, S. J. (2012). There are significant business costs to replacing employees. Center for American Progress. https://www.americanprogress.org/article/there-are-significant-business-costs-to-replacing-employees/

Bureau of Labor Statistics. (2022, September 22). Employee tenure in 2022. https://www.bls.gov/news.release/pdf/tenure.pdf

Cascio, W., & Boudreau, J. W. (2008). Investing in people: Financial impact of human resource initiatives. Prentice-Hall. https://ptgmedia.pearsoncmg.com/images/9780137070923/samplepages/9780137070923.pdf

Chen, B. (2023). Factors of Employee Attrition: A Logistic Regression Approach. https://doi.org/10.54254/2754-1169/20/20230198

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. https://arxiv.org/abs/1603.02754

Cotton, J. L., & Tuttle, J. M. (1986). Employee turnover: A meta-analysis and review with implications for research. Academy of Management Review, 11(1), 55-70. https://doi.org/10.5465/amr.1986.4282625

Daoud, J. I. (2017). Multicollinearity and regression analysis. Journal of Physics: Conference Series, 949, 012009. https://doi.org/10.1088/1742-6596/949/1/012009

De Smet, A., Dowling, B., Mugayar-Baldocchi, M., & Schaninger, B. (2021, September 8). 'Great Attrition' or 'Great Attraction'? The choice is yours. McKinsey Quarterly. https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/great-attrition-or-great-attraction-the-choice-is-yours

Fallucchi, F., Coladangelo, M., Giuliano, R., & De Luca, E. W. (2020). Predicting employee attrition using machine learning techniques. Computers, 9(4), 86. https://doi.org/10.3390/computers9040086

Fortune. (n.d.). Fortune 500. Retrieved December 30, 2022, from https://fortune.com/ranking/fortune500/

Hancock, J. I., Allen, D. G., Bosco, F. A., McDaniel, K. R., & Pierce, C. A. (2013). Meta-analytic review of employee turnover as a predictor of firm performance. Journal of Management, 39(3), 573-603. https://doi.org/10.1177/0149206311424943

Harsha, B. S., Varaprasad, A. J., & Sai Sujith, L. V. N. P. (2020). Early prediction of employee attrition. International Journal of Scientific & Technology Research, 9.

Retrieved from http://www.ijstr.org/final-print/mar2020/Early-Prediction-Of-Employee-Attrition.pdf

Kabir, M. A. A., & Rashedin, R. (2018). Impact of work-life balance on employees' turnover and turnover intentions: An empirical study on multinational corporations in Bangladesh. Jahangirnagar University Journal of Management Research, 1, 15. https://www.researchgate.net/publication/348498931_Impact_of_Work-Life_Balance_on_Employees'_Turnover_and_Turnover_Intentions_An_Empirical_Study_on_Multinational_Corporations_in_Bangladesh

Kaggle. (2017). IBM HR w/more Rows. Kaggle. https://www.kaggle.com/datasets/dgokeeffe/ibm-hr-wmore-rows

Kakad, S., Kadam, R., Deshpande, P., Karde, S., & Lalwani, R. (2020). Employee attrition prediction system. International Journal of Innovative Science, Engineering & Technology, 7(9). ISSN (Online) 2348-7968. https://ijiset.com/vol7/v7s9/IJISET_V7_I9_07.pdf

Kovács, G. (2019). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. Applied Soft Computing. https://doi.org/10.1016/j.asoc.2019.105662

Lam, L., Cheng, B. H., Bamberger, P., & Wong, M.-N. (2022, August 12). Research: The unintended consequences of pay transparency. Harvard Business Review. https://hbr.org/2022/08/research-the-unintended-consequences-of-pay-transparency

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for NAS. https://arxiv.org/pdf/1912.06059.pdf

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Mello, J. A. (2010). Strategic human resource management (3rd ed.). South-Western Cengage Learning. https://books.google.com.sg/books/about/Strategic_Human_Resource_Management.html?id=zkKdQAAACAAJ&redir_esc=y

Mohiuddin, K., Alam, M. A., Alam, M. M., & Welke, P. (2023). Retention Is All You Need. ArXiv. https://arxiv.org/pdf/2304.03103.pdf

Moon, K., Loyalka, P., Bergemann, P., & Cohen, J. (2022). The hidden cost of worker turnover: Attributing product reliability to the turnover of factory workers. Management Science. https://doi.org/10.1287/mnsc.2022.4311

Navarra, K. (2022, April 11). The real costs of recruitment. SHRM. https://www.shrm.org/resourcesandtools/hr-topics/talent-acquisition/pages/the-real-costs-of-recruitment.aspx

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Peterson, S. L. (2004). Toward a theoretical model of employee turnover: A human resource development perspective. Human Resource Development Review, 3(3), 209-227. https://psycnet.apa.org/record/2005-03627-003

Punnoose, R., & Ajit, P. (2016). Prediction of Employee Turnover in Organizations using Machine Learning Algorithms: A case for Extreme Gradient Boosting. (IJARAI) International Journal of Advanced Research in Artificial Intelligence, 5(9), 22. Retrieved from https://pdfs.semanticscholar.org/fa49/19810eaee67e851ad13775b78c94217a7908.pdf

Saeed, F., Mir, A., Hamid, M., & Ayaz, F. (2023). Employee salary and employee turnover intention: A key evaluation considering job satisfaction and job performance as mediators. International Journal of Management Research and Emerging Sciences, 13(1). https://doi.org/10.56536/ijmres.v13i1.234

Shao, Q. (2022). Does less working time improve life satisfaction? Evidence from European Social Survey. Health Economics Review, 12, 50. https://doi.org/10.1186/s13561-022-00396-6

Society for Human Resource Management. (2016). Human capital benchmarking report. https://www.shrm.org/resourcesandtools/business-solutions/documents/human-capital-report-all-industries-all-ftes.pdf

Suresh, A. (2020, November 17). What is a confusion matrix? Analytics Vidhya. https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Tableau. (n.d.). Retrieved November 4, 2023, from https://www.tableau.com/why-tableau/what-is-tableau

Tomaskovic-Devey, D., & Orellana, R. (2022, May 12). The key to retaining young workers? Better onboarding. Harvard Business Review. https://hbr.org/2022/05/the-key-to-retaining-young-workers-better-onboarding

Wei, Y., Yang, R., & Sun, D. (2023). Investigating Tropical Cyclone Rapid Intensification with an Advanced Artificial Intelligence System and Gridded Reanalysis Data. Retrieved from https://www.researchgate.net/publication/367224991_Investigating_Tropical_Cyclone_Rapid_Intensification_with_an_Advanced_Artificial_Intelligence_System_and_Gridded_Reanalysis_Data

Wiles, J. (2021, December 9). Great resignation or not, money won't fix all your talent problems. Gartner. https://www.gartner.com/en/articles/great-resignation-or-not-money-won-t-fix-all-your-talent-problems

Xuecheng, W., Iqbal, Q., & Saina, B. (2022). Factors affecting employee's retention: Integration of situational leadership with social exchange theory. Frontiers in Psychology, 13, 872105. https://doi.org/10.3389/fpsyg.2022.872105

Yoo, W., Mayberry, R., Bae, S., Singh, K., He, Q. (P.), & Lillard, J. W., Jr. (2014). A study of effects of multicollinearity in the multivariable analysis. International Journal of Applied Science and Technology, 4(5), 9–19. PMCID: PMC4318006. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4318006/

Zhao, Y., Hryniewicki, M. K., Cheng, F., Fu, B., & Zhu, X. (2018). Employee Turnover Prediction with Machine Learning: A Reliable Approach. Retrieved from https://www.andrew.cmu.edu/user/yuezhao2/papers/18-intellisys-employee.pdf

Zhu, B., Baesens, B., & Vanthienen, J. (2017). An empirical comparison of techniques for the class imbalance problem in churn prediction. Information Sciences. Retrieved from https://www.semanticscholar.org/paper/An-empirical-comparison-of-techniques-for-the-class-Zhu-Baesens/cffe585aefee3a82fd1e41d00cd22a44eea02824

Appendix

Appendix A: Data dictionary

Note: The data dictionary provided herein was not created by the original curator of the dataset. It has been compiled by the authors of this paper for the purpose of enhancing the understanding and usability of the data for academic research.

No.	Variable	Description
1	Age	The age of the employee in years.
2	Attrition	The status of employee turnover, indicating whether the employee has left the company and the nature of their departure (voluntary, involuntary, etc.).
3	BusinessTravel	The frequency of business-related travel for the employee (e.g., 'Travel_Rarely', 'Travel_Frequently', etc.).
4	DailyRate	The employee's daily wage rate.
5	Department	The department within the company where the employee works (e.g., 'Sales', 'Human Resources', etc.).
6	DistanceFromHome	The distance from the employee's home to the workplace in miles.
7	Education	The level of education of the employee, typically indicated by a numerical scale (e.g., 1 for 'Below College', 2 for 'College', etc.).
8	EducationField	The field of study of the employee's education (e.g., 'Life Sciences', 'Marketing', etc.).
9	EmployeeCount	A count of employees, which is typically a redundant field with a constant value for aggregation purposes.
10	EmployeeNumber	A unique identifier for each employee within the dataset.
11	Application ID	A unique identifier for the employee's job application.

12	EnvironmentSatisfaction	A rating indicating the employee's satisfaction with their work environment.
13	Gender	The gender of the employee (e.g., 'Male', 'Female').
14	HourlyRate	The employee's hourly wage rate.
15	JobInvolvement	A measure of the employee's engagement and involvement in their job.
16	JobLevel	The level or rank of the employee's job within the organization, often on a numerical scale.
17	JobRole	The specific role or title of the employee within the company (e.g., 'Sales Executive', 'Research Scientist', etc.).
18	JobSatisfaction	A rating indicating the employee's satisfaction with their job.
19	MaritalStatus	The marital status of the employee (e.g., 'Single', 'Married', 'Divorced').
20	MonthlyIncome	The employee's monthly salary.
21	MonthlyRate	The monthly rate for the employee, which may include overall compensation beyond base salary.
22	NumCompaniesWorked	The number of companies the employee has worked for prior to the current employer.
23	Over18	An indicator of whether the employee is over 18 years of age (likely a legal requirement check).

24	OverTime	A variable indicating whether the employee works overtime (e.g., 'Yes' or 'No').
25	PercentSalaryHike	The percentage increase in salary the employee has received during the last salary increase cycle.
26	PerformanceRating	The rating of the employee's last performance review.
27	RelationshipSatisfaction	A rating indicating the employee's satisfaction with their relationships at work.
28	StandardHours	The standard number of hours the employee is expected to work, which is typically a fixed value for all employees.
29	StockOptionLevel	The level of stock options granted to the employee, often on a numerical scale.
30	TotalWorkingYears	The total number of years the employee has worked professionally.
31	TrainingTimesLastYear	The number of times the employee received training in the last year.
32	WorkLifeBalance	A rating indicating the employee's perception of the balance between work and personal life.
33	YearsAtCompany	The number of years the employee has been with the current company.
34	YearsInCurrentRole	The number of years the employee has been in their current role within the company.

35	YearsSinceLastPromotion	The number of years since the employee's last promotion.
36	YearsWithCurrManager	The number of years the employee has been working with their current manager.
37	Employee Source	The source from which the employee was recruited (e.g., 'Referral', 'Job Board', etc.).

Appendix B: Hyper-parameter ranges searched through RandomSearchCV

Hyper-parameter range used for Logistic regression model

In the development of the logistic regression model, hyperparameter tuning was performed using a randomized search approach with specific ranges for each hyperparameter. The following table outlines the hyperparameters explored, their definitions, and the respective ranges or sets of values used in the tuning process.

Hyperparameter	Definition	Hyperparameter Range Used
smote__k_neighbors	Number of nearest neighbors to use for SMOTE oversampling	[3, 5, 7, 9, 11]
smote__sampling_strategy	The sampling strategy for SMOTE to balance the dataset	['minority', 'auto', 0.5, 0.75, 1.0]
logistic__C	Inverse of regularization strength for Logistic Regression	Continuous distribution with loc=0, scale=4
logistic__penalty	The norm used in the penalization for Logistic Regression	['l2', 'none']

Hyperparameter Range Used for XGBoost Classifier

The optimization of the XGBoost classifier involved an extensive search over a predefined range of hyperparameters. The table below lists the hyperparameters that were included in the randomized search, their corresponding descriptions, and the range of values that were evaluated.

Hyperparameter	Definition	Hyperparameter Range Used
smote__k_neighbors	Number of nearest neighbors to use with SMOTE	[3, 5, 7, 9, 11, 13]
smote__sampling_strategy	SMOTE strategy to balance the dataset	['minority', 'auto', 0.3, 0.5, 0.75]
xgboost__learning_rate	Step size shrinkage used in update to prevent overfitting	Continuous distribution from 0.01 to 0.31
xgboost__max_depth	Maximum depth of a tree	[3, 5, 7]
xgboost__n_estimators	Number of trees to fit	[100, 250, 500]
xgboost__subsample	Subsample ratio of the training instances	Continuous distribution from 0.6 to 1
xgboost__colsample_bytree	Subsample ratio of columns when constructing each tree	Continuous distribution from 0.6 to 1

Appendix C: Optimal Hyper-parameters found for Predictive Models

The hyperparameter tuning process identified a set of optimal hyperparameters for both the logistic regression model and the XGBoost classifier. These parameters were determined using a randomized search with cross-validation. Below are the best hyperparameter values that yielded the most effective model performance.

Logistic Regression Model Optimal Hyperparameters:

Hyperparameter	**Optimal Value**
logistic__C	1.2035132392670786
logistic__penalty	'l2'
smote__k_neighbors	3
smote__sampling_strategy	0.5

The optimal hyperparameters for the logistic regression model include a regularization strength (logistic__C) of approximately 1.20, L2 penalty, a SMOTE k-neighbors value of 3, and a SMOTE sampling strategy of 0.5.

XGBoost Classifier Optimal Hyperparameters:

Hyperparameter	**Optimal Value**
smote__k_neighbors	13
smote__sampling_strategy	'minority'
xgboost__colsample_bytree	0.6421977039321082

xgboost__learning_rate	0.14696037114487306
xgboost__max_depth	5
xgboost__n_estimators	250
xgboost__subsample	0.8157368967662603

For the XGBoost classifier, the optimal set of hyperparameters includes a SMOTE k-neighbors value of 13, a SMOTE sampling strategy targeting the minority class, a colsample_bytree ratio of approximately 0.64, a learning rate of approximately 0.15, a max depth of 5, 250 estimators, and a subsample ratio of approximately 0.82.

Appendix D: Guide to setup Tableau dashboard functionality, which requires TabPy and Jupyter Notebook

This appendix serves as a guide for Windows users to install and configure TabPy, which allows users to execute Python scripts and leverage machine learning models within Tableau. This guide assumes Python, Jupyter Notebook and relevant software like Tableau have already been installed. The following steps outline the process of setting up TabPy on a Windows environment.

Step 1: Install TabPy Using pip

- Open the Command Prompt as an administrator. You can do this by searching for cmd in the Start menu, right-clicking on Command Prompt, and selecting "Run as administrator."
- Install TabPy by executing the following command:

  pip install tabpy

- Wait for the installation to complete. pip will handle the installation of TabPy and its dependencies.

Step 2: Start the TabPy Server

After installing TabPy, you can start the server by running the following command in the Command Prompt:

tabpy

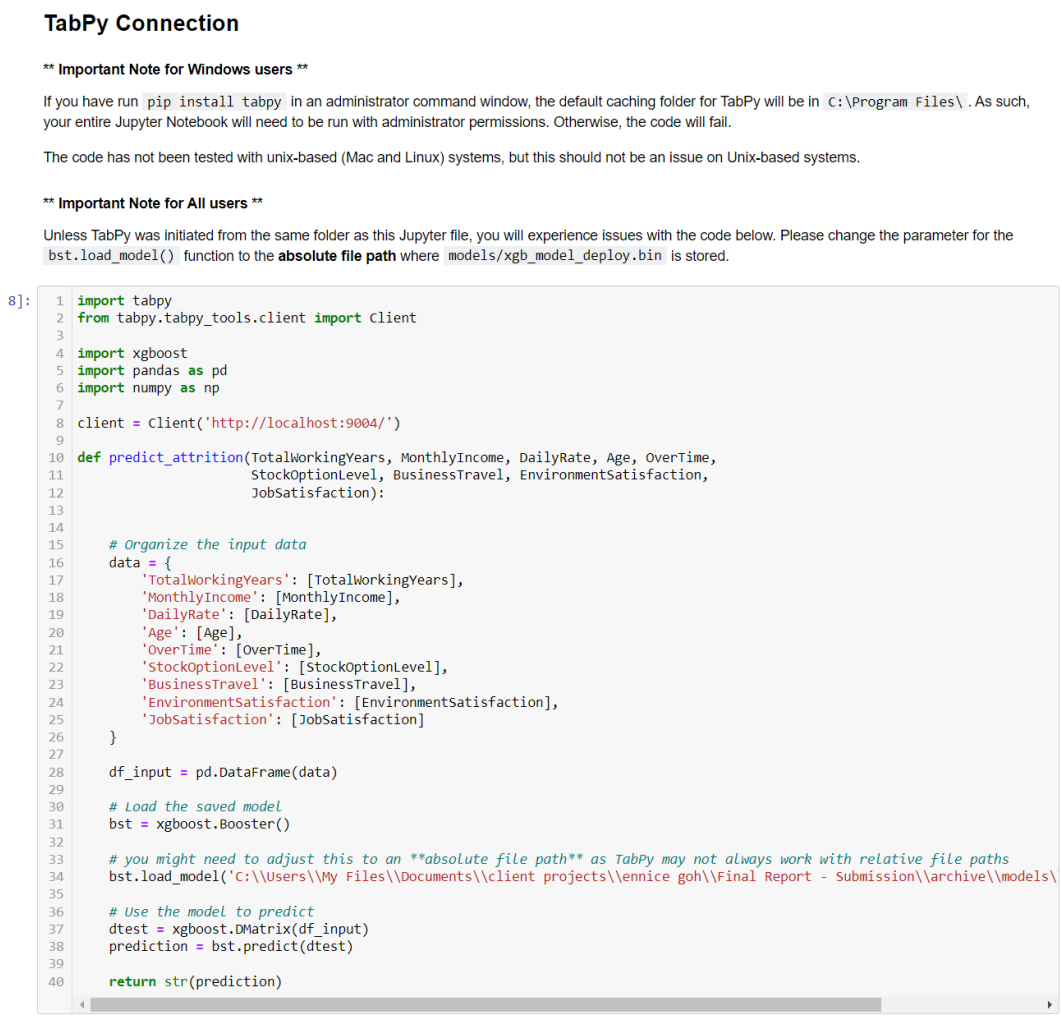
This command starts the TabPy server on the default port, which is 9004. If you wish to run it on a different port, use the --port argument.

Step 3: Configure Tableau to Use TabPy

- Open Tableau Desktop.
- Navigate to "Help" > "Settings and Performance" > "Manage External Service Connection" to configure Tableau to use TabPy.
- Select 'TabPy/External API' and enter the server's address. If you're running TabPy locally, this will be localhost or 127.0.0.1.
- Enter the port number where TabPy is running, which by default is 9004.
- Click 'Test Connection' to ensure that Tableau can communicate with TabPy.
- Once the connection is successful, click 'OK' to save the settings.

Step 4: Run Jupyter Notebook code to connect to client

Open EDA, Modelling, Evaluation, Interpretation code.ipynb and scroll all the way down to the section header **TabPy Connection.**



TabPy Connection

** Important Note for Windows users **

If you have run `pip install tabpy` in an administrator command window, the default caching folder for TabPy will be in `C:\Program Files\`. As such, your entire Jupyter Notebook will need to be run with administrator permissions. Otherwise, the code will fail.

The code has not been tested with unix-based (Mac and Linux) systems, but this should not be an issue on Unix-based systems.

** Important Note for All users **

Unless TabPy was initiated from the same folder as this Jupyter file, you will experience issues with the code below. Please change the parameter for the `bst.load_model()` function to the **absolute file path** where `models/xgb_model_deploy.bin` is stored.

8]:

```
import tabpy
from tabpy.tabpy_tools.client import Client

import xgboost
import pandas as pd
import numpy as np

client = Client('http://localhost:9004/')

def predict_attrition(TotalWorkingYears, MonthlyIncome, DailyRate, Age, OverTime,
                      StockOptionLevel, BusinessTravel, EnvironmentSatisfaction,
                      JobSatisfaction):


    # Organize the input data
    data = {
        'TotalWorkingYears': [TotalWorkingYears],
        'MonthlyIncome': [MonthlyIncome],
        'DailyRate': [DailyRate],
        'Age': [Age],
        'OverTime': [OverTime],
        'StockOptionLevel': [StockOptionLevel],
        'BusinessTravel': [BusinessTravel],
        'EnvironmentSatisfaction': [EnvironmentSatisfaction],
        'JobSatisfaction': [JobSatisfaction]
    }

    df_input = pd.DataFrame(data)

    # Load the saved model
    bst = xgboost.Booster()

    # you might need to adjust this to an **absolute file path** as TabPy may not always work with relative file paths
    bst.load_model('C:\\Users\\My Files\\Documents\\client projects\\ennice goh\\Final Report - Submission\\archive\\models\

    # Use the model to predict
    dtest = xgboost.DMatrix(df_input)
    prediction = bst.predict(dtest)

    return str(prediction)
```

Without running the rest of the code, just execute all the cells here until the end of the page. This connects to the TabPy client.

Then, you can just open the Tableau dashboard and it should work. You may need to adjust one of the sliders for the model to run and display its prediction.

Note:

Make sure your firewall and/or antivirus settings allow communication over the port you've configured for TabPy.

Troubleshooting:

If you encounter any issues, check the following:

- Ensure that the Python executable is in your PATH.
- Verify that all necessary ports are open and not blocked by a firewall.

- Make sure that TabPy is running before trying to connect from Tableau.

Potential known issues on Windows:

If you have run `pip install tabpy` in an administrator command window, the default caching folder for TabPy will be in `C:\Program Files\`. As such, your entire Jupyter Notebook will need to be run with administrator permissions. Otherwise, the code will fail.

The code has not been tested with Unix-based (Mac and Linux) systems.

Important Note for All users:

Unless TabPy was initiated from the same folder as this Jupyter file, you will experience issues with the code below. Please change the parameter for the `bst.load_model()` function to the absolute file path where `models/xgb_model_deploy.bin` is stored.

Appendix E: Python Environment Used to Build Predictive Models

This appendix provides details on the Python environment and the libraries used to develop and run the machine learning models presented in this study.

Hardware Specifications:

- Processor: AMD Ryzen 7 5800H
- RAM: 64GB
- Operating System: Windows 10 64-bit

Software Environment:

- Programming Language: Python 3.9.5
- Development Environment: Jupyter Notebook
- Key Python Libraries:
    - **scikit-learn**: Comprehensive machine learning library used for implementing various ML algorithms (version 1.1.0).
    - **imbalanced-learn**: Integrated with scikit-learn, this library provides a range of re-sampling techniques to address imbalanced datasets (version 0.11.0).
    - **xgboost**: An optimized gradient boosting library designed to be highly efficient, flexible, and portable (version 1.7.6).

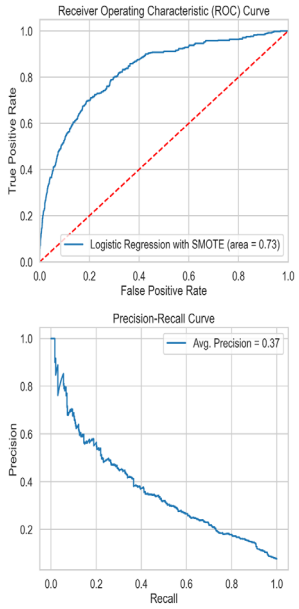
Code Repository: The Python code utilized for running the models is maintained in a Jupyter Notebook environment, ensuring an interactive and iterative approach to data analysis and model development. The code comprises scripts for data preprocessing, model training, hyperparameter tuning, model validation, and performance evaluation, utilizing the libraries mentioned above.

Appendix F: Python Code used to build predictive models and perform model interpretation with SHAP

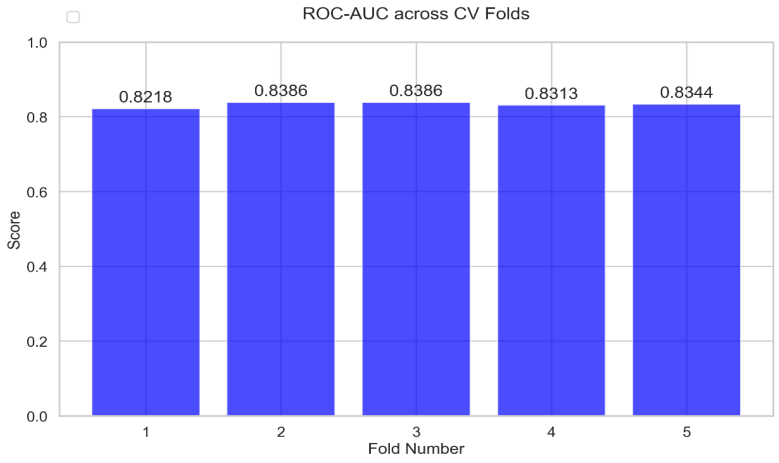
Python Code used to build and perform hyper-parameter tuning with Logistic regression model

```
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(drop='first', handle_unknown='ignore'),
df.drop(columns='Attrition').select_dtypes(include=['object']).columns.tolis
t())  # OneHotEncoder with drop one
    ],
    remainder=StandardScaler()
)

# Create a pipeline with SMOTE and Logistic Regression using imblearn's
Pipeline
pipeline = ImbPipeline([
    ('preprocessor', preprocessor),
    ('smote', SMOTE(random_state=42, n_jobs=-1)),
    ('logistic', LogisticRegression(solver='newton-cg', max_iter=10000,
n_jobs=-1, random_state=42))
])

param_dist = {
    'smote__k_neighbors': [3, 5, 7, 9, 11],
    'smote__sampling_strategy': ['minority','auto', 0.5, 0.75, 1.0],
    'logistic__C': uniform(loc=0, scale=4),
    'logistic__penalty': ['l2', 'none']
}

rs = RandomizedSearchCV(
    pipeline,
    param_distributions=param_dist,
    scoring='roc_auc',
    n_iter=50,
    cv=5,
    verbose=10,
    random_state=42,
    n_jobs=-1
)

X = df.drop(columns='Attrition')
y = df['Attrition']

y = y.map({'Attrition': 1, 'Non-Attrition': 0})

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
shuffle=True, random_state=42, stratify=y)

rs.fit(X_train, y_train)

print("Best Hyperparameters:", rs.best_params_)

y_pred = rs.best_estimator_.predict(X_test)
```

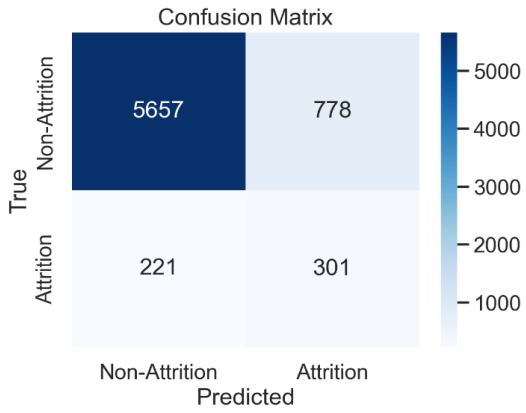
Logistic regrssion: ROC curve and Precision-recall curve



Logistic regression: ROC-AUC across CV folds plot



Logistic regression: Confusion matrix



Coefficients of Logistic regression model

0	BusinessTravel_Travel_Frequently	1.668
1	BusinessTravel_Travel_Rarely	1.044
44	TotalWorkingYears	-0.950
20	OverTime_Yes	0.843
9	Gender_Male	0.637
27	Employee Source_Referral	0.578
16	JobRole_Sales Executive	-0.535
11	JobRole_Laboratory Technician	0.531
19	MaritalStatus_Single	0.491
7	EducationField_Other	-0.481
4	EducationField_Life Sciences	-0.414
10	JobRole_Human Resources	0.399
12	JobRole_Manager	-0.379
2	Department_Research & Development	-0.360
17	JobRole_Sales Representative	0.351
28	Employee Source_Seek	-0.308
29	Age	-0.302
33	EnvironmentSatisfaction	-0.298
6	EducationField_Medical	-0.285
35	JobInvolvement	-0.284
39	NumCompaniesWorked	0.270
26	Employee Source_Recruit.net	-0.257
22	Employee Source_GlassDoor	-0.254
36	JobSatisfaction	-0.246
25	Employee Source_LinkedIn	-0.211

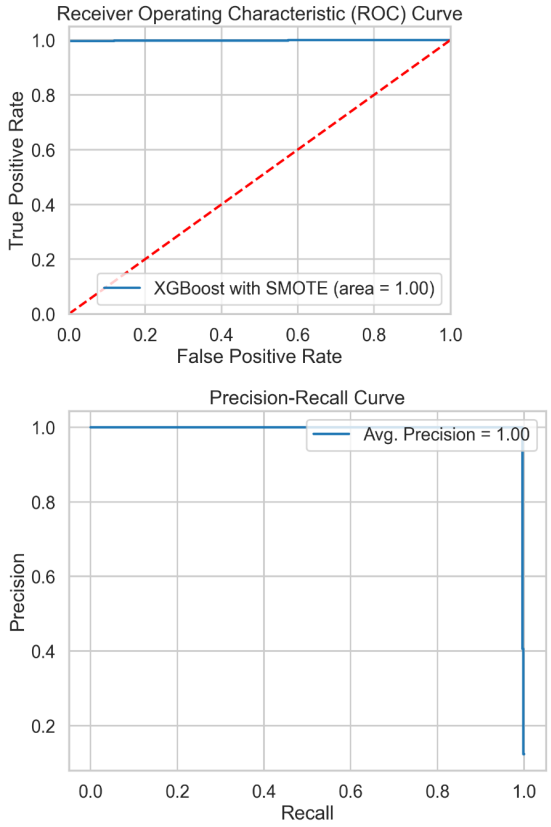
Python Code used to build, perform hyper-parameter tuning and save XGBoost Classifier model:

```
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(drop='first', handle_unknown='ignore'),
df.drop(columns='Attrition').select_dtypes(include=['object']).columns.tolis
t())  # OneHotEncoder with drop one
    ],
    remainder='passthrough'
)
```

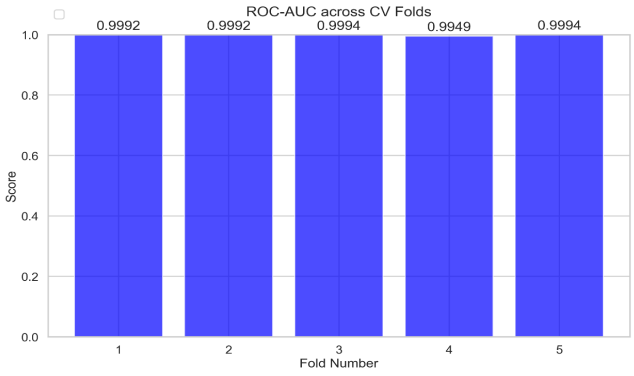
```
# Create a pipeline with SMOTE and XGBClassifier using imblearn's Pipeline
pipeline = ImbPipeline([
    ('preprocessor', preprocessor),
    ('smote', SMOTE(random_state=42, n_jobs=-1)),
    ('xgboost', XGBClassifier(eval_metric='logloss'))
])

# Hyperparameter grid (you can adjust this as per your needs for XGBoost)
param_dist = {
    'smote__k_neighbors': [3, 5, 7, 9, 11, 13],
    'smote__sampling_strategy': ['minority', 'auto', 0.3, 0.5, 0.75],
    'xgboost__learning_rate': uniform(0.01, 0.3),
    'xgboost__max_depth': [3, 5, 7],
    'xgboost__n_estimators': [100, 250, 500],
    'xgboost__subsample': uniform(0.6, 0.4),
    'xgboost__colsample_bytree': uniform(0.6, 0.4)
}

# Initialize RandomSearch
rs = RandomizedSearchCV(
    pipeline,
    param_distributions=param_dist,
    scoring='roc_auc',
    n_iter=50,
    cv=5,
    verbose=1,
    random_state=42,
    n_jobs=-1
)

X = df.drop(columns='Attrition')
y = df['Attrition']

y = y.map({'Attrition': 1, 'Non-Attrition': 0})

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
shuffle=True, random_state=42, stratify=y)

rs.fit(X_train, y_train)

print("Best Hyperparameters:", rs.best_params_)

# Use the best estimator pipeline to predict
y_pred = rs.best_estimator_.predict(X_test)

# Save model for future use
xgb_model = rs.best_estimator_.named_steps['xgboost']

try:
    directory = 'models'

    if not os.path.exists(directory):
        os.makedirs(directory)

    filename = f"xgb_model_{datetime.now().strftime('%Y%m%d_%H%M%S')}.bin"
    xgb_model.save_model(f'{directory}/{filename}')
    print(f'Model saved in {directory}/{filename}')
except:
```

```
print('Model has been trained but unable to save')
```

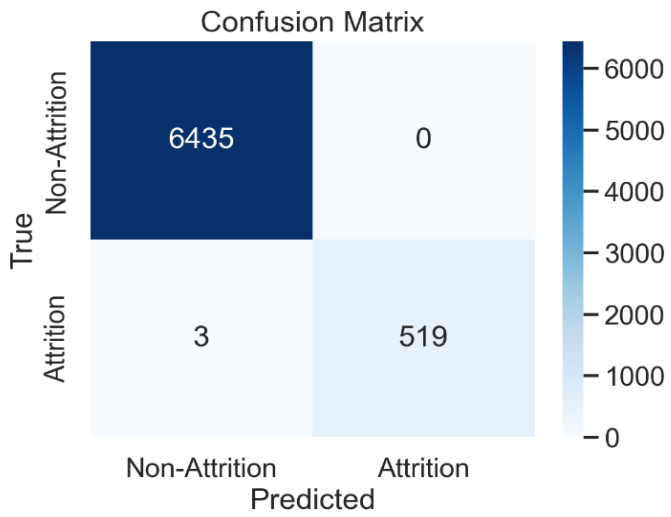
XGBoost Classifier: ROC curve and Precision-recall curve



XGBoost Classifier: ROC-AUC across CV folds plot



XGBoost Classifier: Confusion matrix


Confusion Matrix
6435
0
3
519
Non-Attrition
Attrition
True
Non-Attrition
Attrition
Predicted
6000
5000
4000
3000
2000
1000
0